

画像処理ライブラリ



WIL-Bui lder チュートリアル (Ver3.0.0 以降対応)

☆第1版☆

(株)ファーストは、同社が提供していない装置における同社製ソフトウェア・ハードウェアの使用または信頼性についてはいかなる責任も負いません。(株)ファーストは本書で記載されているソフトウェア・ハードウェアの内容、商品価値、又は特定の使用目的に対する責任に対して明示又は黙示に関わらずいかなる保証も行いません。

本書の内容は、予告なしに変更することがあります。内容の変更について、(株)ファーストはいかなる責任も負いません。本書あるいは関連ソフトウェアにおける誤りから生じる損害について、(株)ファーストはいかなる責任も負いません。

本書の内容の一部または全部を転載することは固くお断りします。

御注意

©Microsoft, Windows, Visual Studio, Visual C++, Visual C#, Visual Basic は、米国 Microsoft Corporation の登録商標です。

©.NET は、米国 Microsoft Corporation の商標です。

©Windows XP は、米国 Microsoft Corporation の商品名です。

©Windows Vista は、米国 Microsoft Corporation の登録商標です。

©Windows 7 は、米国 Microsoft Corporation の登録商標です。

©Intel, MMX, Pentium は、米国 Intel Corporation の登録商標です。

©その他、文中における会社名、商品名は各社の登録商標または商標です。

©文中では 商標シンボル(®)、(TM)、(C) 等の表記は省略致します。

1 . はじめにお読みください	1
1.1 本ドキュメントについて	1
1.2 関連ドキュメントについて	1
2 . 基本操作編	2
2.1 画像を入力してみよう	2
2.1.1 カメラからの画像入力	3
2.1.2 ファイルからの画像入力	5
2.2 簡単な画像処理を試みよう	7
2.3 処理範囲を設定してみよう	10
3 . 実践編	12
3.1 2値ブローブ解析を設定してみよう	12
3.1.1 2値ブローブ解析	12
3.1.2 結果の表示等	14
3.2 グレイサーチを設定してみよう	17
3.2.1 パターン登録	17
3.2.2 パターンの設定	19
3.2.3 サーチ実行	20
3.2.4 結果の表示等	22
3.3 ハフ検出を設定してみよう	25
3.3.1 エッジを抽出しよう。	25
3.3.2 抽出したエッジデータでハフ検出	27
3.3.3 2直線の交点	28
4 . 応用編	32
4.1 配列走査について	32
4.2 ループ処理について	33
4.3 条件分岐について	34
4.4 プラグイン機能について	41
5 . サンプルワークフロー解説	42
5.1 SAMPLE00	42
5.2 SAMPLE01	43
5.3 SAMPLE02	44
5.4 SAMPLE03	45
5.5 SAMPLE04	46
5.6 SAMPLE05	47
5.7 SAMPLE06	48
5.8 SAMPLE07	49
5.9 SAMPLE08	50
5.10 SAMPLE09	51
5.11 SAMPLE10	52

目 次

5.12 SAMPLE11	53
6 . 困ったときは	54
6.1 トラブルシューティング	54
6.2 ユーザ・サポートについて	55

1. はじめにお読みください

この度は、FAST Vision Library シリーズ WIL をご購入頂きまして誠にありがとうございます。

1.1 本ドキュメントについて

本書では「WIL-Builder 操作説明書」により、基本操作を理解している方を前提に、WIL-Builder で画像処理を構築する方法を解説しています。

1.2 関連ドキュメントについて

WIL-Builder で画像処理を構築していく場合、Function の機能説明は、以下の関連ドキュメントをご参照ください。

① FVCL Reference(ヘルプ)

WILのライブラリリファレンス (FVCL版) とプログラマーズガイドを統合したものです。
WILセットアップ時にインストールされるヘルプファイルにてご提供いたします。

② FVIL Reference(ヘルプ)

WILのライブラリリファレンス (FVIL版) とプログラマーズガイドを統合したものです。
WILセットアップ時にインストールされるヘルプファイルにてご提供いたします。

③ FIE説明書(ヘルプ)

WILのFIEライブラリのリファレンスです。
WILセットアップ時にインストールされるヘルプファイルにてご提供いたします。

④ 各種ボードの取扱説明書

ファースト製各種ボードのハードウェアに関する情報が記載されています。ファースト製各種ボードをご購入いただきましたお客様を対象とした取扱説明書です。

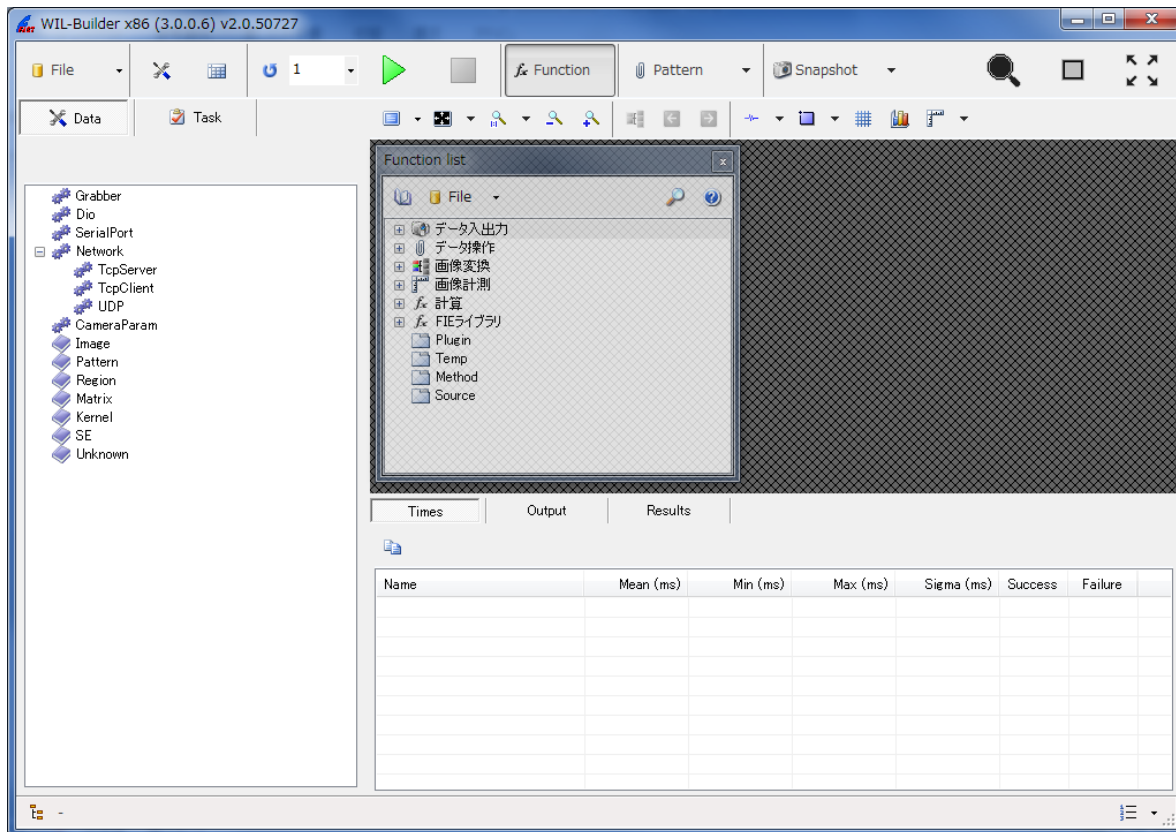
⑤ 画像処理解説書

WIL に搭載されている各画像処理手法の解説書です。アルゴリズムやパラメータについて解説しています。

2. 基本操作編

WIL-Builder の基本的な操作の流れを説明します。

起動時の画面。



2.1 画像を入力してみよう

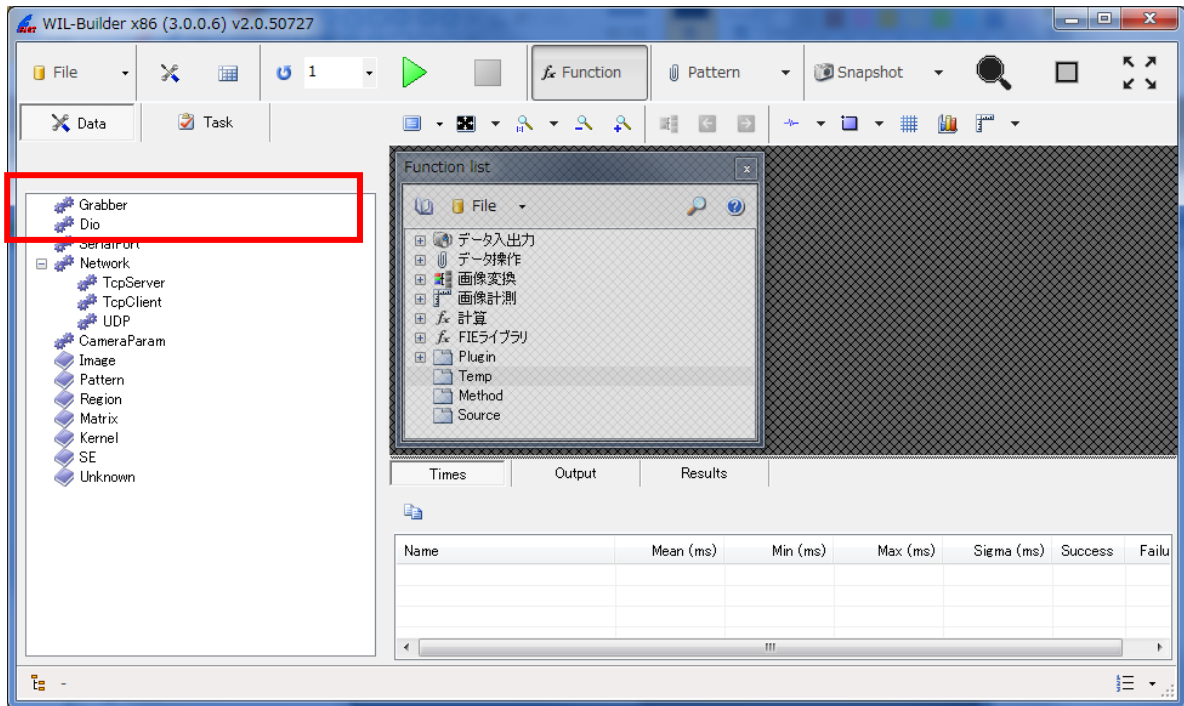
カメラからの画像入力と、画像ファイルを読み込んだの画像入力の手順を説明します。

2.1.1 カメラからの画像入力

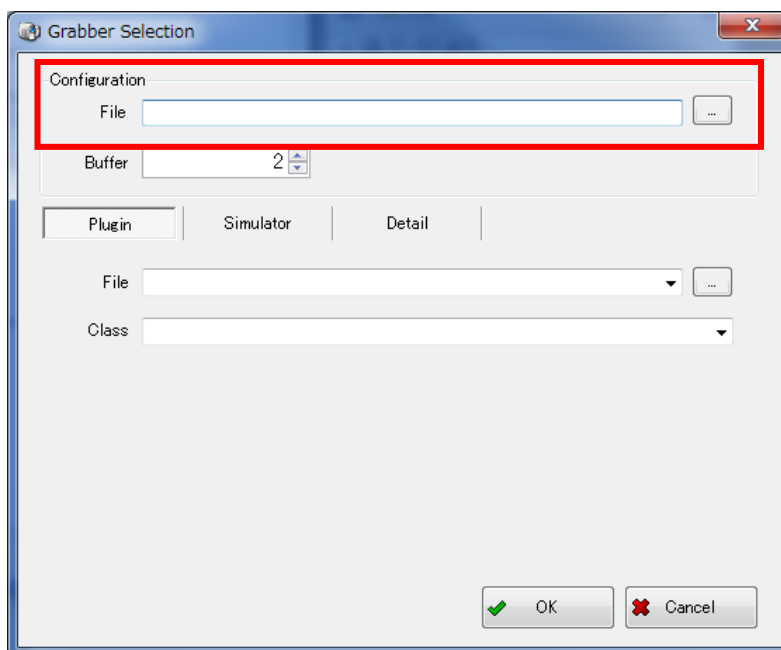
WIL では、Grabber Board 毎に対応しているカメラ用の「カメラ設定ファイル(ini ファイル)」が用意されています。

カメラから画像入力を行う場合、この ini ファイルを選択する必要があります。

- ① Data タブの「Grabber」を右クリックし、Open を選択します。



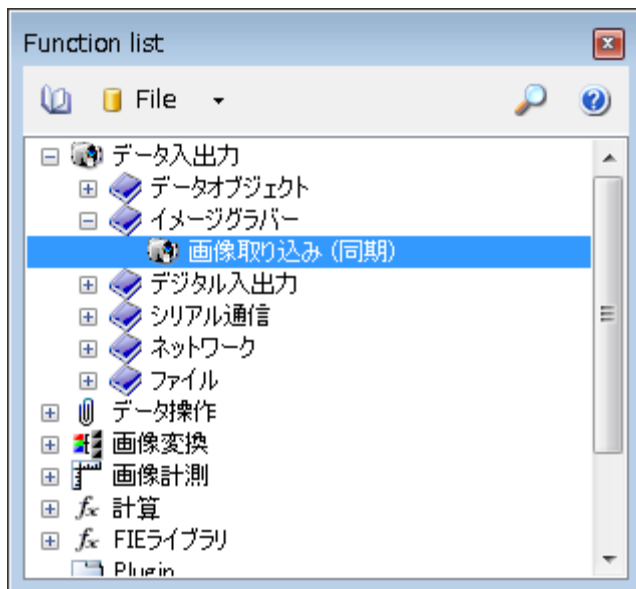
- ② Grabber Selection ダイアログが表示されますので、Configuration の File から使用するカメラ用の ini ファイルを選択します。選択後「OK」をクリックして、ダイアログを閉じます。



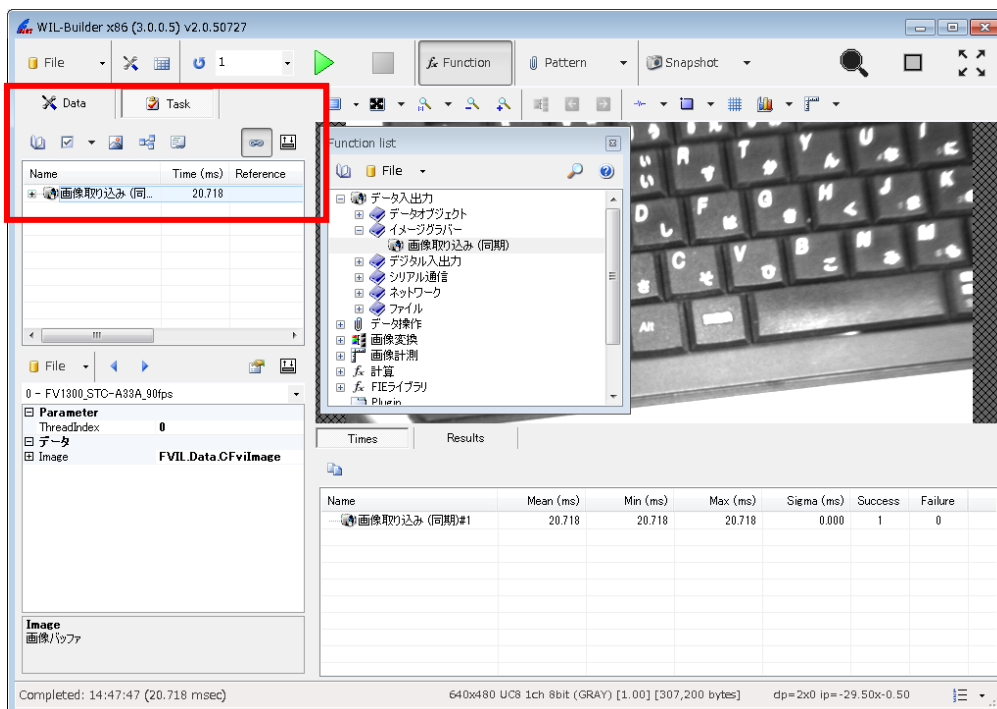
- ③ ツールバーの「Function」をクリックし、Function list をアクティブにします。



- ④ Function list の「データ入出力」、「イメージグラバー」の順にリストを展開し「画像取り込み」をダブルクリックします。



- ⑤ TASK タブのワークフローリストに「画像取り込み」が追加されます。

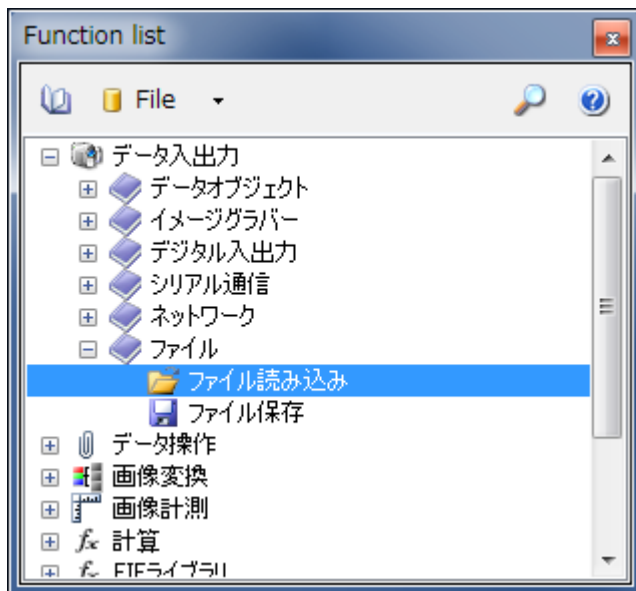


- ⑥ 画像ビューにカメラから取り込んだ画像が表示されます。

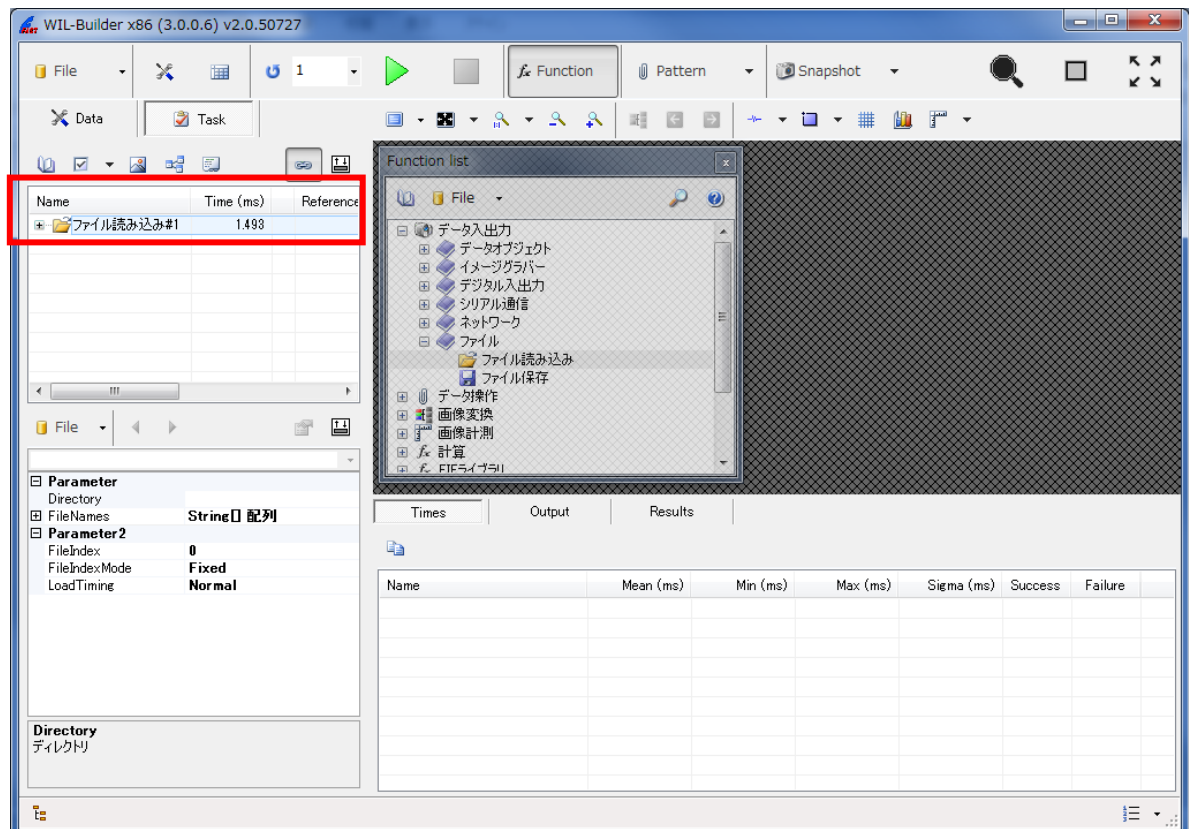
2.1.2 ファイルからの画像入力

カメラからの画像入力ではなく、画像ファイルを読み込んで、画像処理を行う事も可能です。
手順を説明します。



- ① Function list の「データ入出力」の+マークをクリックします。
リストが展開されますので、同様に「ファイル」の+マークをクリックします。
さらにリストが展開されますので「ファイル読み込み」をダブルクリックします。

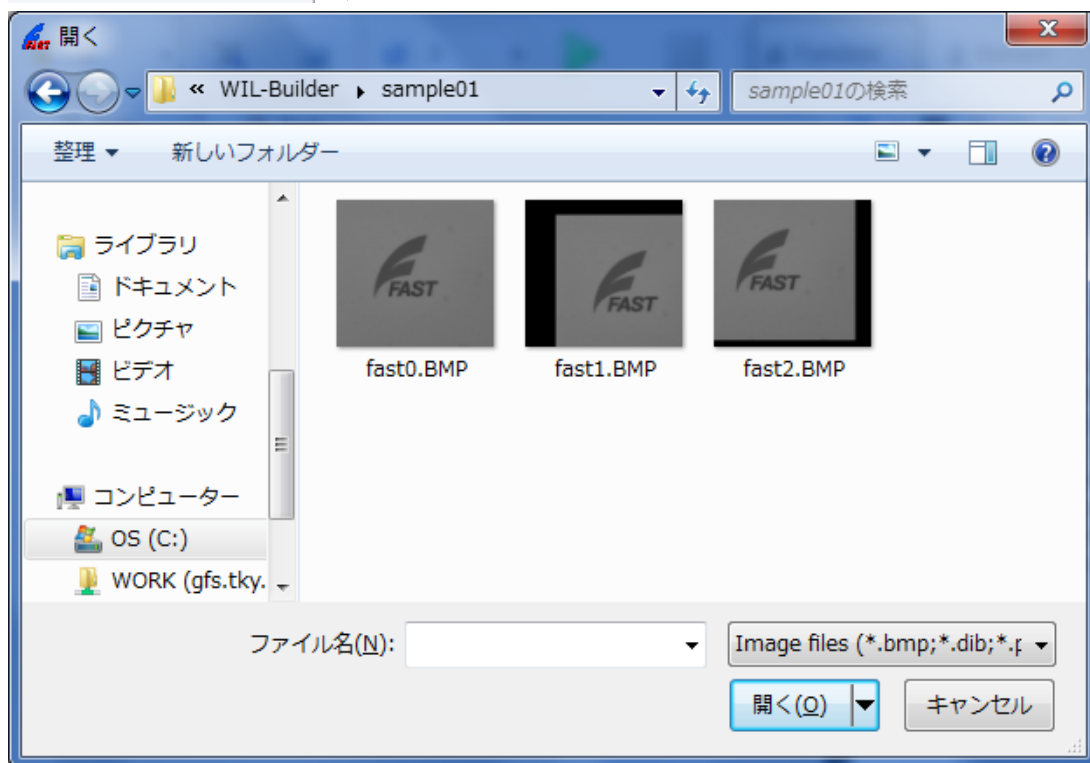


- ② TASK タブのワークフローリストに「ファイル読み込み」が追加されます。



- ③ TASK タブの下側にプロパティグリッドが表示されていますので、「FileNames」をクリックし、画像ファイルを指定します。

String[] 配列   ここをクリックすると画像ファイル名を選択できます。



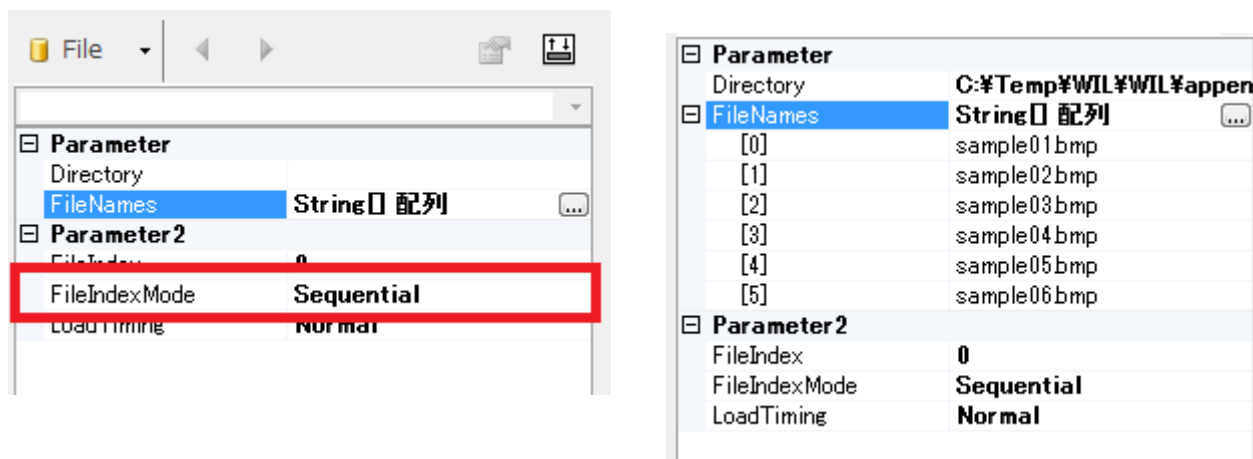
※インストール時に指定したサンプルのフォルダを指定します。

デフォルトのフォルダは「デスクトップ¥WIL¥appendix¥Samples¥WIL-Builder」となっています。

※対象ファイルフォーマットは、BMP、JPEG、PNG、Raw、TIFF となっています。

- ④ ③で指定した画像ファイルが、画像ビューに表示されます。

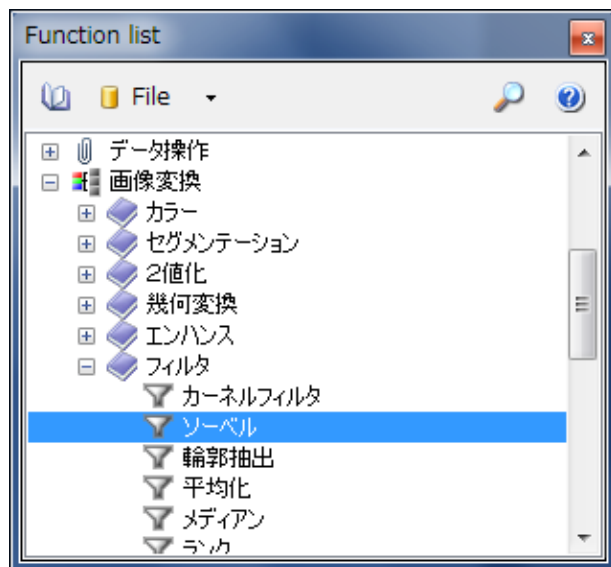
※複数の画像ファイルを設定することも可能です。「ファイル読み込み」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter2 の FileIndexMode を Fixed から Sequential に変更し、上記のファイル選択時に Ctrl キーを押しながら、ファイル名を複数選択することができます。



2.2 簡単な画像処理をしてみよう

画像が入力できるようになったので、簡単な画像処理をしてみましょう。

- ① 画像が入力できるようになった TASK に、画像処理を追加します。
- ② Function list の「画像変換」、「フィルタ」の順にリストを展開し、「ソーベル」をダブルクリックします。

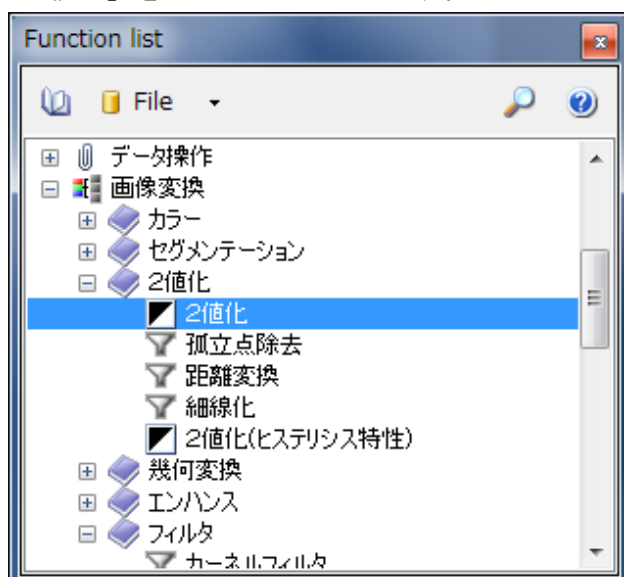


- ③ TASK タブのワークフローリストに「ソーベル」が追加され、画像ビューに変換された画像が表示されます。(FVIL.Filter.CFviSobelDir)

Name	Time (ms)	Reference
+ ファイル読み込み#1	1.172	
+ ソーベル#1	0.169	



- ④ 変換された画像を2値化します。Function listの「画像変換」、「2値化」の順にリストを展開し、「2値化」をダブルクリックします。

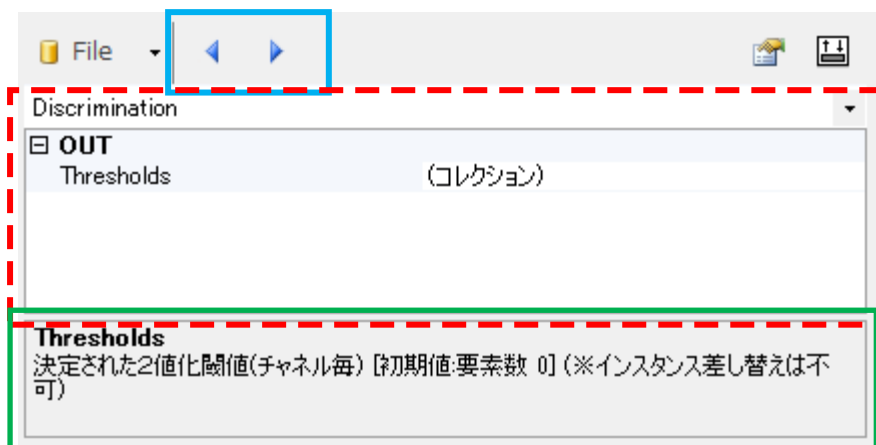


- ⑤ TASK タブのワークフローリストに「2 値化」が追加され、画像ビューに、2 値化された画像が表示されます。(FVIL. Conversion. CFviBinarizeDiscrimination)

Name	Time (ms)	Reference
ファイル読み込み#1	0.834	
ソーベル#1	6.325	
2値化#1	48.892	



複数の機能を持つノードを選択すると青いカーソル(下図青枠)とリストボックス(下図橙枠)が有効になります。これらで機能の指標を切り替えることができます。青いカーソルは機能の指標をシーケンスに増減します。(左:1つ減少、右:1つ増加) リストボックスは機能の指標と識別名の一覧を表示しています



※対象ブローブが2値化できる最適なノードを選択してください。

ノードの切り替えは、上図緑枠を確認してください。現在設定されているノードが表示されます。

※画像データの受け渡しは、自動的に行っていますが、任意に設定することも可能です。

出力ファイル(Out or Dst)を入力ファイル(Src)へドラッグアンドドロップすることにより、受け渡しを行うことができます。

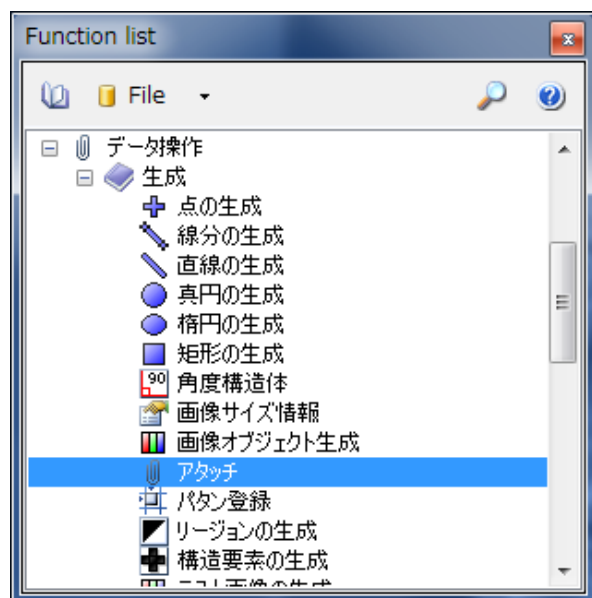
受け渡しの修正方法は、ドラッグをやり直すか、右クリックし、「Reset」を選択してください。

Data			Task
Name	Time (ms)	Reference	
ファイル読み込み#1	60.990		
← FileIndex			
→ Out			
→ Directory			
→ FileName			
→ FileIndex			
ソーベル#1	0.482	ファイル読み込み#1.Out	
← Src0			
→ Dst0			
2値化#1	0.148	ソーベル#1.Dst0	
← Src0			
→ Dst0			

2.3 処理範囲を設定してみよう

処理範囲を設定してみましょう。処理範囲の設定は、画像メモリを管理するクラス (FVIL. Data. CFvi Image) で行います。

- ① 画像処理が行えるようになった TASK に、処理範囲を追加します。
- ② Function list の「データ操作」、「生成」の順にリストを展開し、「アタッチ」をダブルクリックします。



- ③ TASK タブのワークフローリストの最後の行に「アタッチ」が追加されます。

Name	Time (ms)	Reference
ファイル読み込み#1	1.872	
ソーベル#1	10.957	
2値化#1	0.518	
アタッチ#1	1.345	

(FVIL. Data. CFvi Image. Window)

- ④ 「アタッチ」を、ドラッグして「ファイル読み込み」の後ろに移動します。

Name	Time (ms)	Reference
ファイル読み込み#1	1.872	
アタッチ#1	1.345	
ソーベル#1	10.957	
Src0		ファイル読み込み#1
Dst0		
2値化#1	0.518	

- ⑤ アタッチを移動した為、画像データの受け渡しが、崩れたので、ドラッグアンドドロップで修正します。

Name	Time (ms)	Reference
ファイル読み込み#1	1.872	
FileIndex		
Out		
Directory		
FileName		
FileIndex		
アタッチ#1	0.016	ファイル読み込み#1.Out
Image		
Window		
Image		
ソーベル#1	0.329	アタッチ#1.Image
Src0		
Dst0		
2値化#1	0.518	

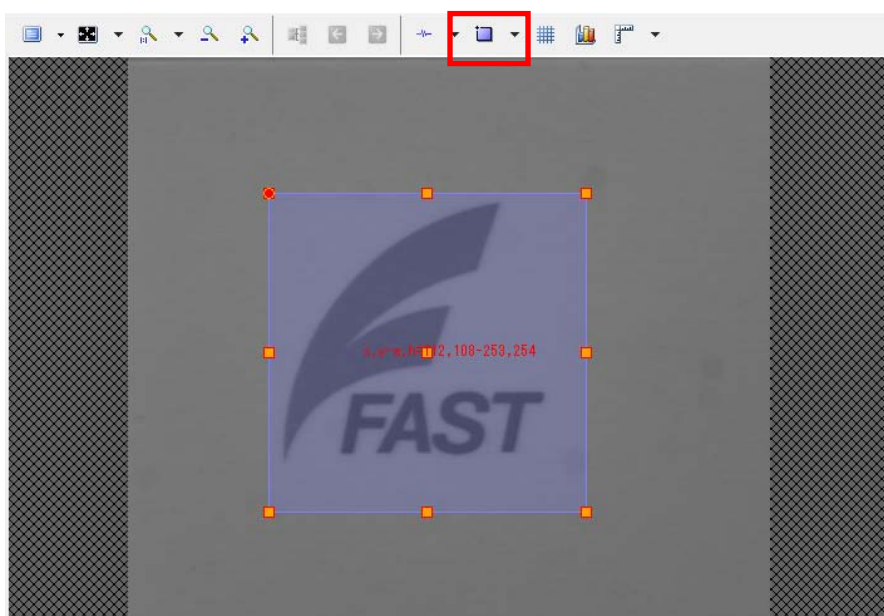
- ⑥ 「アタッチ」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「Data」をクリックすると Parameter が展開されます。一番下の「Window」が処理範囲となっています。

データ	
Data	FVIL.Data.CFviiImage
Bpp	8
Channel	1
Depth	8
HorzSize	512
ImageInfo	GRAY
ImageType	UC8
Offset	0, 0
VertSize	480
Window	112, 108, 253, 254

X 始点、Y 始点、X サイズ、Y サイズの順です。

- ⑦ 「Window」の値を変更すると、画像処理を行う範囲が変更されます。

※TASK タブのワークフローリスト内の「アタッチ」をクリックした状態で、ツールバーの WOI メニューをクリックすると処理範囲をマウスで設定することもできます。設定終了後、再度 WOI メニューをクリックするとマウス設定モードが解除されます。



3. 実践編

基本的な操作が理解できたので、実践的な操作をしてみましょう。

3.1 2 値ブローブ解析を設定してみよう

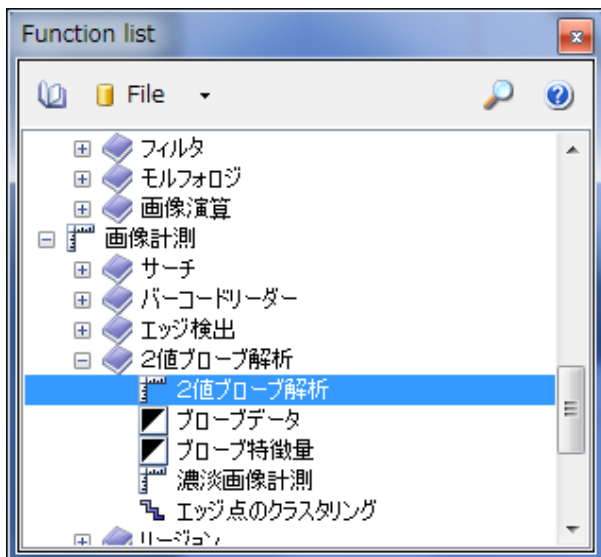
2 値化した画像に対しブローブ解析を設定します。

なお、画像入力や 2 値化、処理範囲の設定は基本操作編で説明したので省略します。

3.1.1 2 値ブローブ解析

2 値化した画像から、選択した色のブローブを解析する機能 (FVIL Blob) を設定します。

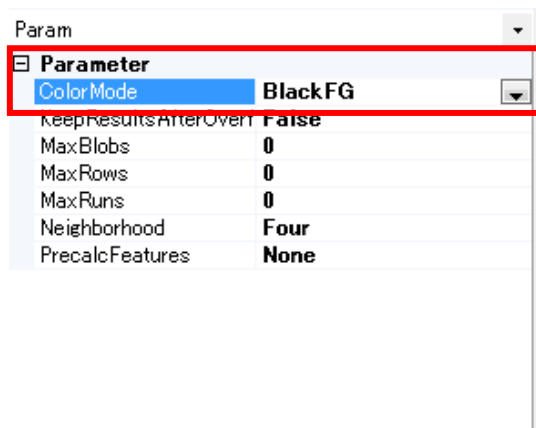
- ① Function list の「画像計測」、「2 値ブローブ解析」の順にリストを展開し、「2 値ブローブ解析」をダブルクリックします。



- ② TASK タブのワークフローリストの最後の行に「2 値ブローブ解析」が追加されます。
(FVIL Blob, CFviBlob)

Name	Time (ms)	Reference
+ ファイル読み込み#1	2.472	
+ アタッチ#1	2.204	
+ 2値化#1	0.116	
+ 2値ブローブ解析#1	12.901	+

- ③ 「2 値ブローブ解析」をクリックすると、TASK タブの下側にプロパティグリッドが表示されていますので、「ColorMode」をクリックし、解析したい色を選択できます。



- ④ 画像ビューに選択された色のブローブが解析され、表示されます。

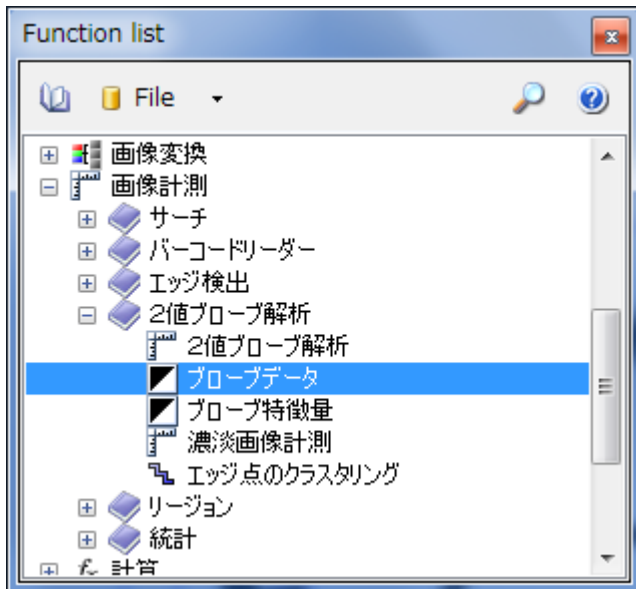


※この画像は SAMPLE01 フォルダの FAST0.BMP の画像に対し、Threshold を 80 に設定し、Black を対象に解析したものです。

3.1.2 結果の表示等

2 値ブローブ解析で得られた結果を使用して、2 点間の距離を計測します。

- ① Function list の「画像計測」、「2 値ブローブ解析」の順にリストを展開し、「ブローブデータ」をダブルクリックします。

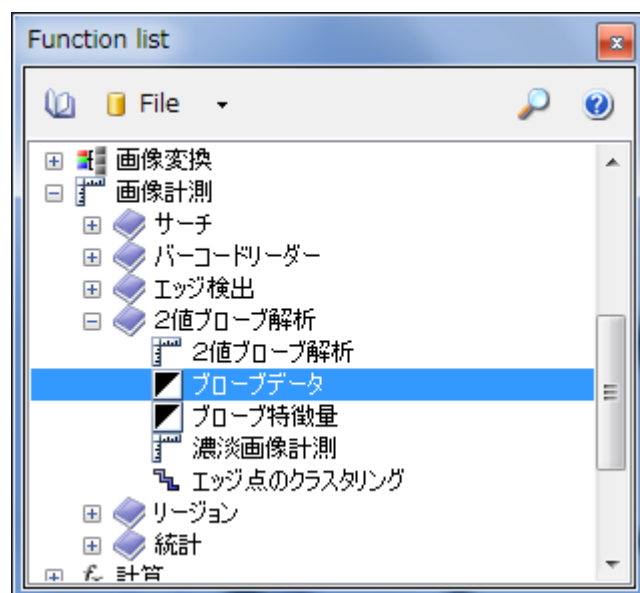


- ② TASK タブのワークフローリストの最後の行に「ブローブデータ」が追加されます。

Name	Time (ms)	Reference
ファイル読み込み#1	0.980	
アタッチ#1	0.016	
2値化#1	0.532	
2値ブローブ解析#1	1.671	+
ブローブデータ#1	0.013	

(FVIL. Blob. CFviBlobData)

- ③ さらにもう一つ「ブローブデータ」を追加します。



- ④ TASK タブのワークフローリストの最後の行に「ブローブデータ」が追加されます。

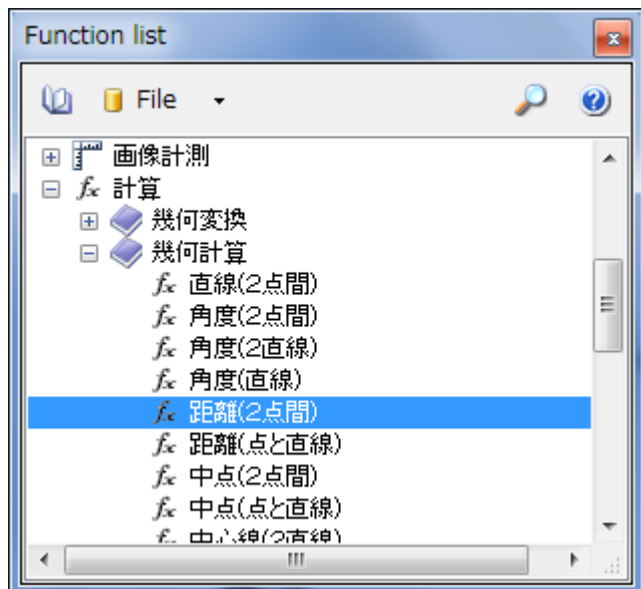
Name	Time (ms)	Reference
ファイル読み込み#1	0.980	
アタッチ#1	0.016	
2値化#1	0.532	
2値ブローブ解析#1	1.671	+
ブローブデータ#1	0.011	
ブローブデータ#2	0.021	

- ⑤ 「ブローブデータ#2」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「Index」を「1」に変更します。

Parameter	
Index	1
データ	
Data	FVIL.Blob.CFviBlobData

※画面内に複数個のブローブがあることを前提に、設定しています。

- ⑥ Function list の「計算」、「幾何計算」の順にリストを展開し、「距離 (2 点間)」をダブルクリックします。



- ⑦ TASK タブのワークフローリストの最後の行に「距離 (2 点間)」が追加されます。

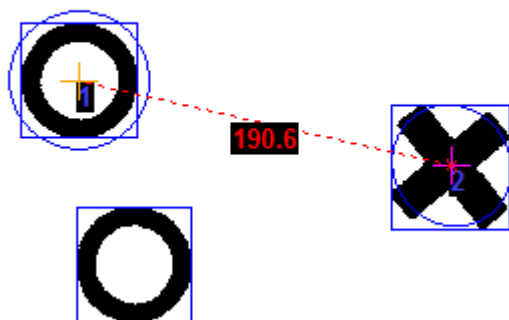
Name	Time (ms)	Reference
ファイル読み込み#1	0.786	
アタッチ#1	0.007	
2値化#1	2.015	
2値ブローブ解析#1	0.105	+
ブローブデータ#1	0.035	
ブローブデータ#2	0.005	
距離(2点間)#1	0.085	
Point0		ブローブデータ#2.Center
Point1		ブローブデータ#1.Center
Distance		

(FVIL.Caliper.Function.Distance)

- ⑧ 実行アイコンをクリックしてみましょう



画像ビューに 2 点間を計測した値が、画素で表示されます。



3.2 グレイサーチを設定してみよう

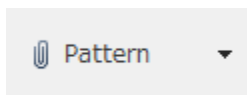
濃淡画像でパタンを登録し、そのパタンを処理範囲の中からサーチする機能を設定します。
なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します。

3.2.1 パタン登録

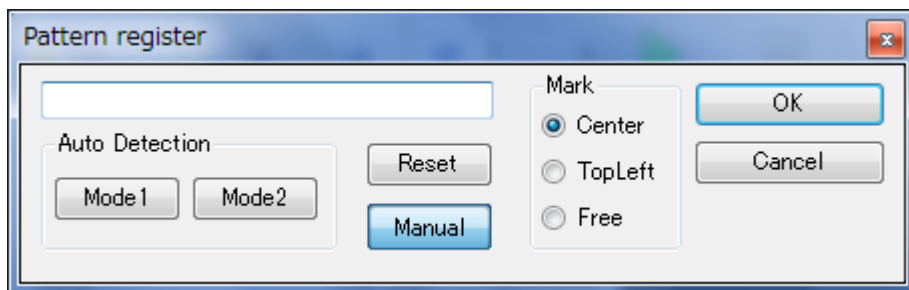
パタンの登録は、ワークフロー内ではなく、メイン画面のツールバーにある「パタン登録」をクリックすることで登録することができます。

※登録方法の詳細「WIL-Builder 説明書」参照してください。

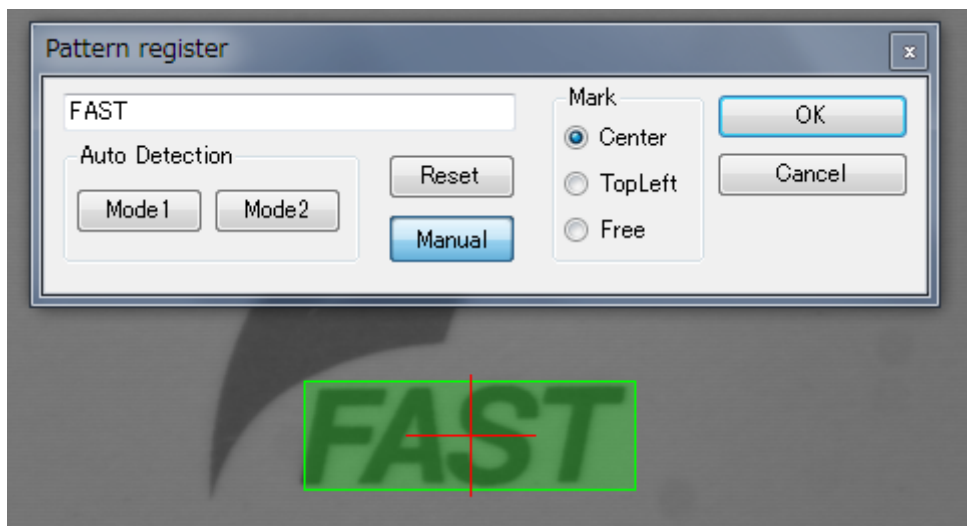
- ① メイン画面のツールバーにある「パタン登録」をクリックします。



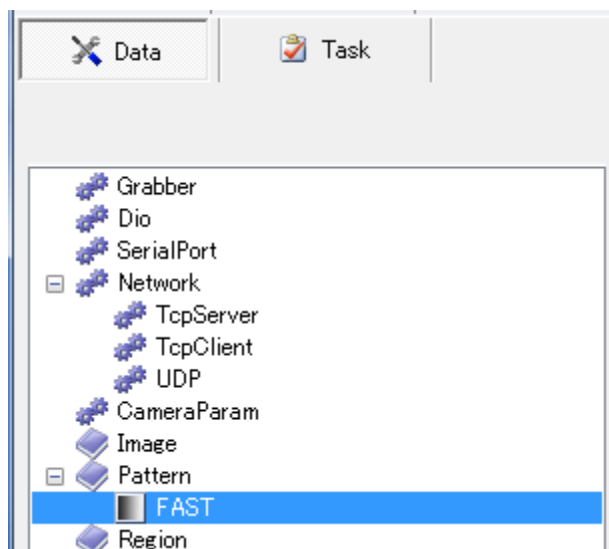
- ② ツールボックスが表示されます



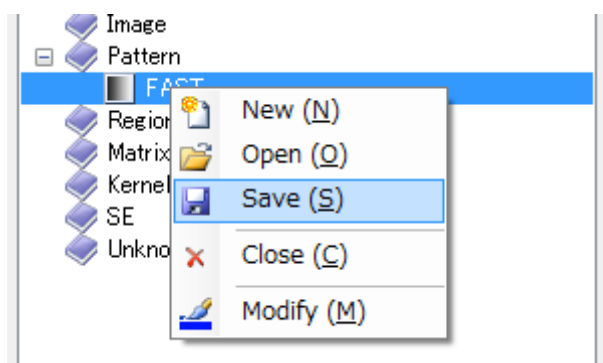
- ③ 登録したい領域をマウスで設定し、パタン名称を記入し登録します。



- ④ 登録されたパターンは、DATA タブの「Pattern」に表示されます。



- ⑤ 登録したパターンを保存する場合は、パターン名称を右クリックし、Save を選択します。



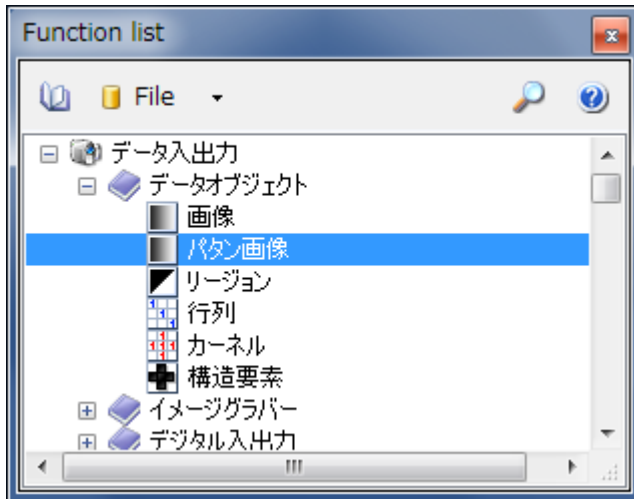
3.2.2 パタンの設定

サーチを行う為のワークフローを設定します。

メイン画面のツールバーにある「パターン登録」をクリックして、パターンは登録したので、それを使用するための手続きを設定します。

なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します

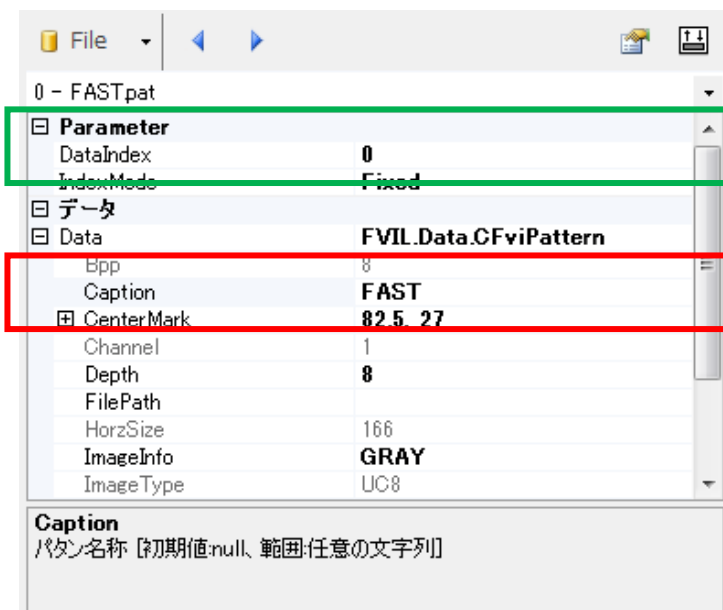
- ① Function list の「データ入出力」、「データオブジェクト」の順にリストを展開し、「パターン画像」をダブルクリックします。



- ② TASK タブのワークフローリストの最後の行に「パターン画像」が追加されます。

Name	Time (ms)	Reference
ファイル読み込み#1	1.227	
パターン画像#1	0.004	

Auto Link によりパターンが関連付けされます。(FVIL.Data.CFviPattern)

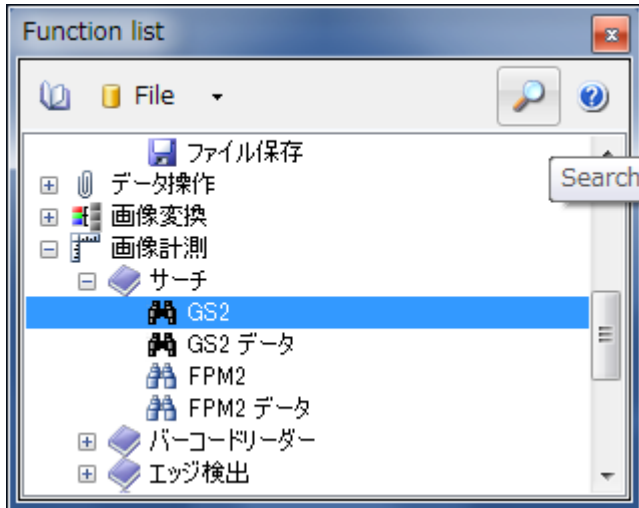


Data の Caption に先ほど登録したパターン名称がない場合、またパタンの名称を変更したい場合は、Caption へパタンの名称を入力してください。(左図赤枠)

複数パターンを登録した場合は、登録順に Index が割り振られているので Parameter の DataIndex を変更することでパターンが変更されます。(左図緑枠)

3.2.3 サーチ実行

- ① Function list の「画像計測」、「サーチ」の順にリストを展開し、「GS2」をダブルクリックします。



- ② TASK タブのワークフローリストの最後の行に「GS2」が追加されます。

Name	Time (ms)	Reference
+ ファイル読み込み#1	1.035	
+ パタン画像#1	0.002	
+ GS2#1	2.604	+

(FVIL. GS2. CFviGS2)

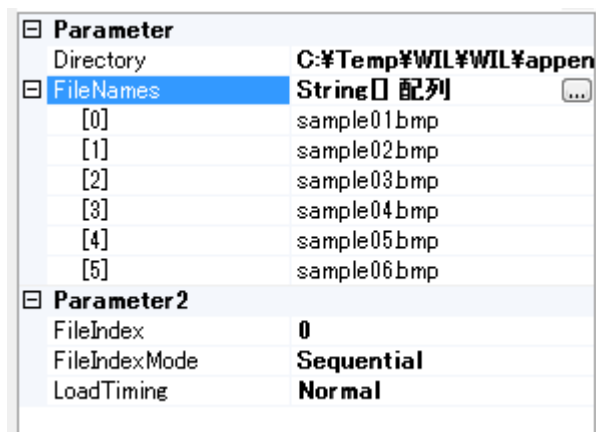
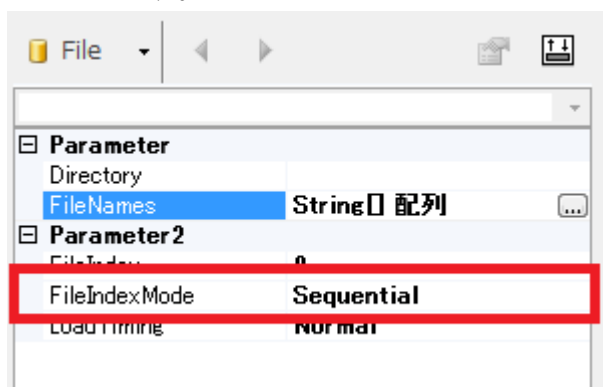
- ③ 実行アイコンをクリックしてみましょう。



画像ビューに登録したパタンをサーチした画像が表示されます。



※複数の画像ファイルを設定することも可能です。「ファイル読み込み」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter2 の FileIndexMode を Fixed から Sequential に変更し、上記のファイル選択時に Ctrl キーを押しながら、ファイル名を複数選択することができます。

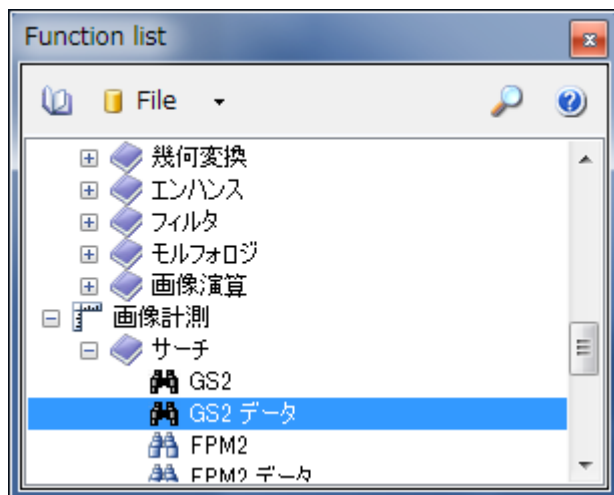


3.2.4 結果の表示等

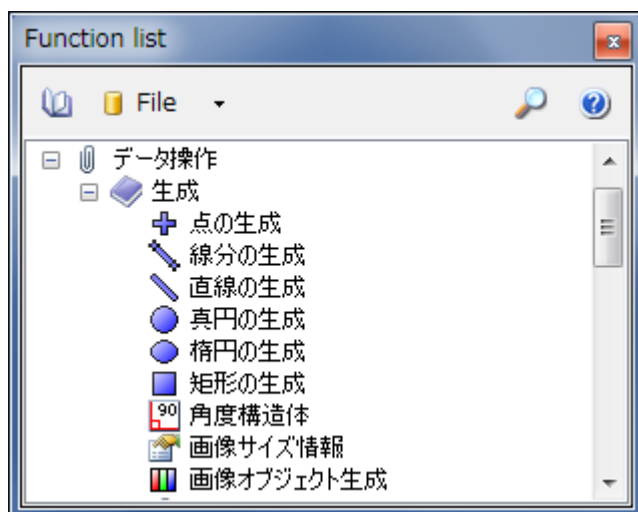
グレイサーチで得られた結果を使用して、任意の点との2点間の距離を計測します。

「GS2」から単一のデータを取得するため、「GS2 データ」を設定します。

- ① Function list の「画像計測」、「サーチ」の順にリストを展開し、「GS2 データ」をダブルクリックします。



- ② 続けて、Function list の「データ操作」、「生成」の順にリストを展開し、「点の生成」をダブルクリックします。

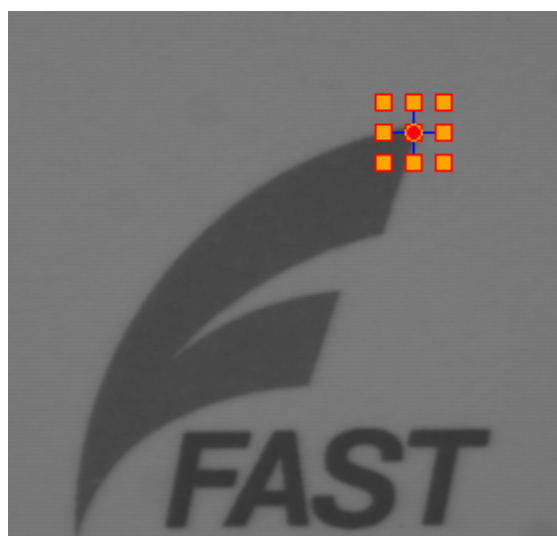


- ③ TASK タブのワークフローリストの最後の行に「点の生成」が追加されます。

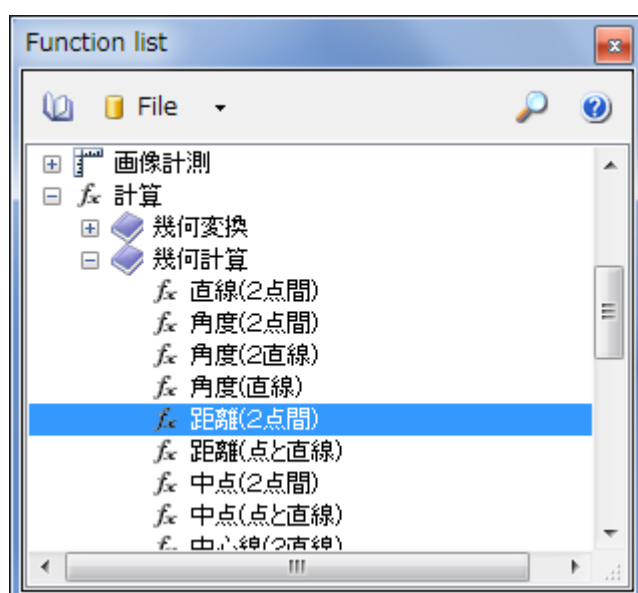
Name	Time (ms)	Reference
+ ファイル読み込み#1	1.048	
+ パターン画像#1	0.001	
+ GS2#1	1.265	+
+ GS2 データ#1	0.010	
+ 点の生成#1	0.078	

(FVIL. GDI. CFviGdiPoint)

- ④ ワークフローリストの「点の生成」をクリックすると、点の位置をマウスで設定することができますので、任意の位置に設定します。



- ⑤ Function list の「計算」、「幾何計算」の順にリストを展開し、「距離 (2 点間)」をダブルクリックします。



- ⑥ TASK タブのワークフローリストの最後の行に「距離 (2 点間)」が追加されます。

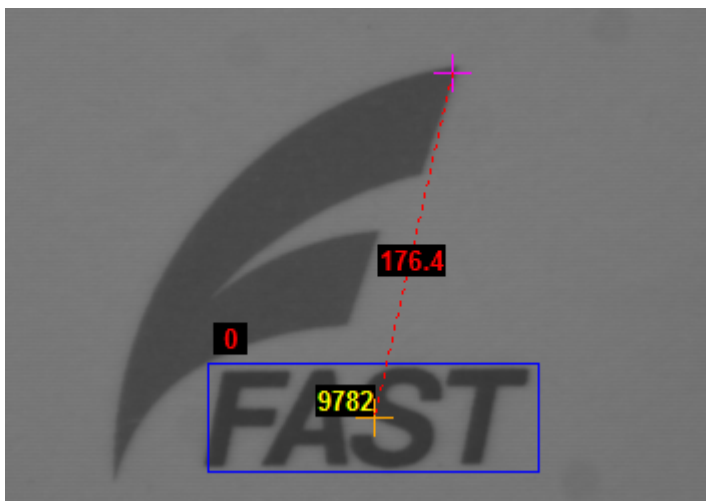
Name	Time (ms)	Reference
ファイル読み込み#1	1.048	
ボタン画像#1	0.001	
GS2#1	1.265	+
GS2 データ#1	0.008	
点の生成#1	0.078	
距離(2点間)#1	0.032	
Point0		点の生成#1.this
Point1		GS2 データ#1.Position
Distance		

(FVIL.Caliper.Function.Distance)

- ⑦ 実行アイコンをクリックしてみましょう



- ⑧ 画像ビューに 2 点間を計測した値が、画素で表示されます。

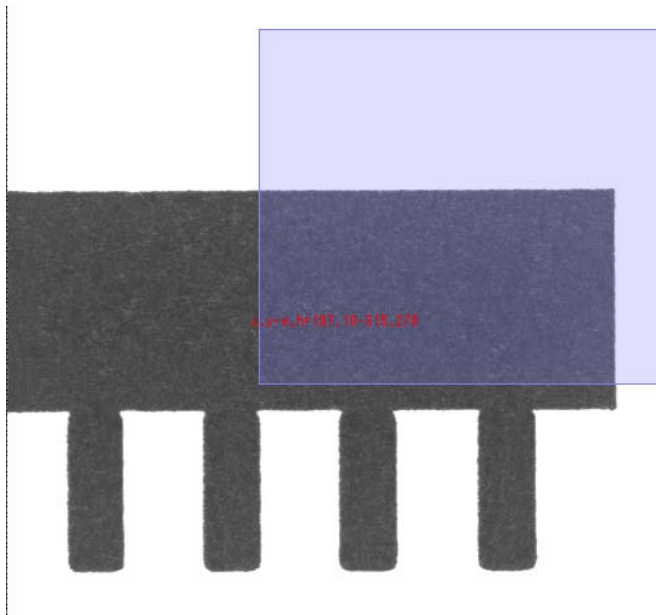


3.3 ハフ検出を設定してみよう

濃淡画像でエッジを抽出し、ハフ変換で直線を検出する機能を設定します。

なお、画像入力や処理範囲の設定は基本操作編で説明しましたので省略します。

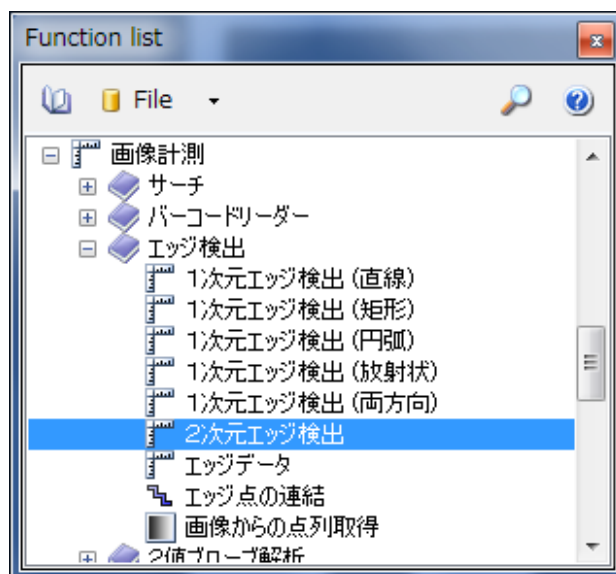
使用する画像は、「sample05¥007.bmp」を使用し、処理範囲も設定します。



3.3.1 エッジを抽出しよう。

濃淡画像から、エッジを抽出します。

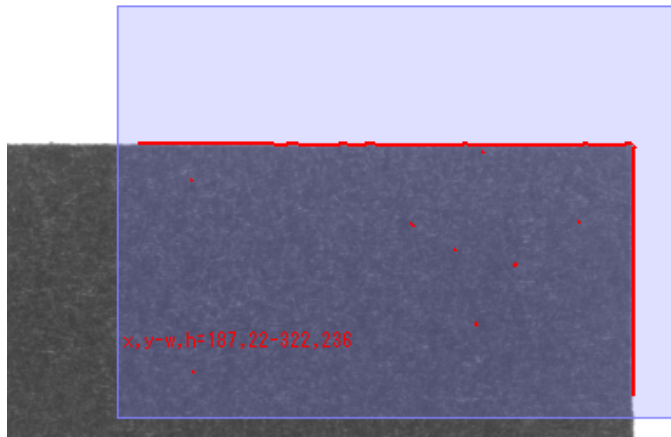
- ① Function list の「画像計測」、「エッジ検出」の順にリストを展開し、「2次元エッジ検出」をダブルクリックします。



- ② TASK タブのワークフローリストの最後の行に「2次元エッジ抽出」が追加されます。

Name	Time (ms)	Reference
+ ファイル読み込み#1	3.948	
+ アタッチ#1	0.018	
+ 2次元エッジ検出#1	5.371	+

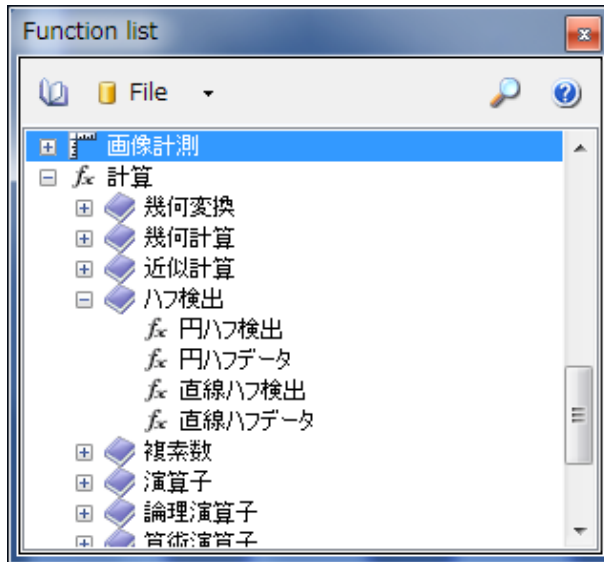
(FVIL. Edge. CFviEdge2D)



3.3.2 抽出したエッジデータでハフ検出

抽出されたエッジデータでハフ検出を設定し、実行します。

- ① Function list の「計算」、「ハフ検出」の順にリストを展開し、「直線ハフ検出」をダブルクリックします。



- ② TASK タブのワークフローリストの最後の行に「直線ハフ検出」が追加されます。

Name	Time (ms)	Reference
+ ファイル読み込み#1	0.793	
+ 2次元エッジ検出#1	14.702	+
+ 直線ハフ検出#1	0.989	

(FVIL. Hough. FunctionEx. LineHough)

- ③ 実行アイコンをクリックしてみましょう



- ④ 画像ビューに検出された直線が表示されます。



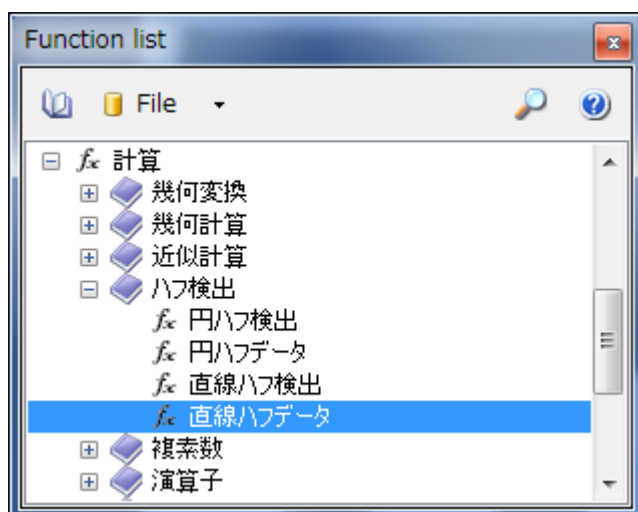
3.3.3 2 直線の交点

ハフ検出で直線を 2 本検出し、その交点を算出する設定をします。

- ① 「直線ハフ検出」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「RequestNum」を「2」に変更します。



- ② Function list の「計算」、「ハフ検出」の順にリストを展開し、「直線ハフデータ」をダブルクリックします。

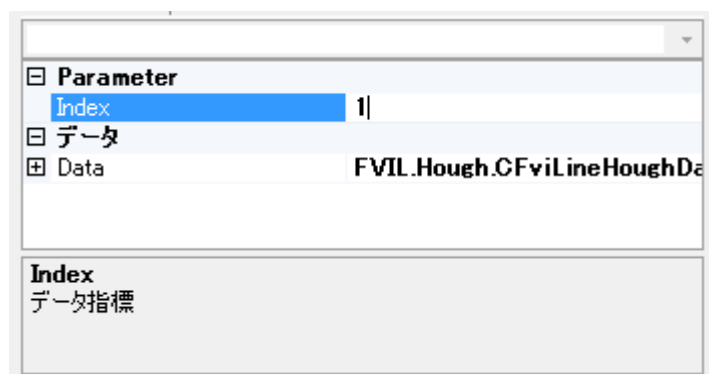


- ③ さらにもう一つ「直線ハフデータ」を追加します。
- ④ TASK タブのワークフローリストの最後の行に「直線ハフデータ」が追加されます。

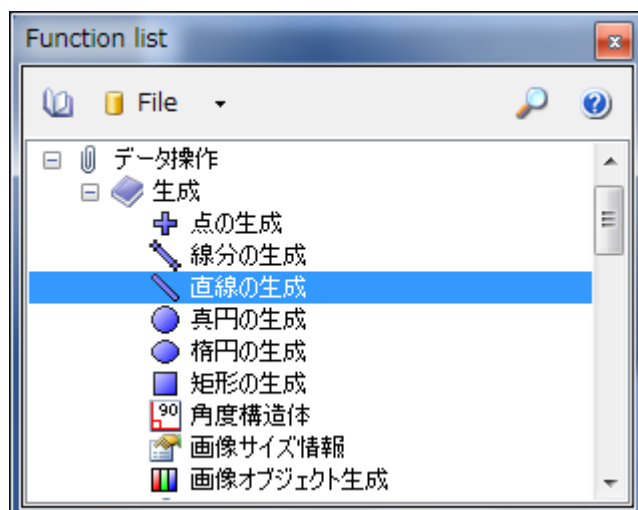
Name	Time (ms)	Reference
ファイル読み込み#1	0.811	
2次元エッジ検出#1	7.486	+
fx 直線ハフ検出#1	0.876	
fx 直線ハフデータ#1	0.004	
fx 直線ハフデータ#2	0.011	

(FVIL. Hough. CFviLineHoughData)

- ⑤ 「直線ハフデータ#2」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「Index」を「1」に変更します。



- ⑥ Function list の「データ操作」、「生成」の順にリストを展開し、「直線の生成」をダブルクリックします。



- ⑦ さらにもう一つ「直線の生成」を追加します。

- ⑧ TASK タブのワークフローリストの最後の行に「直線の生成」が追加されます。

Name	Time (ms)	Reference
+ ファイル読み込み#1	0.811	
+ 2次元エッジ検出#1	7.486	+
+ 直線ハフ検出#1	0.876	
+ 直線ハフデータ#1	0.004	
+ 直線ハフデータ#2	0.016	
+ 直線の生成#1	0.003	
+ 直線の生成#2	0.002	

(FVIL.GDI. CFviGdiLine)

- ⑨ 「直線ハフデータ#1」の値を「直線の生成#1」の値とリンクします。

Name	Time (ms)	Reference
直線ハフデータ#1	0.004	
Result		直線ハフ検出#1.Datas
Index		
this		
a		
b		
c		
q		
Score		
直線ハフデータ#2	0.016	
直線の生成#1	0.003	直線ハフデータ#1a 直線ハフデータ#1b 直線ハフデータ#1c
a		
b		
c		
this		

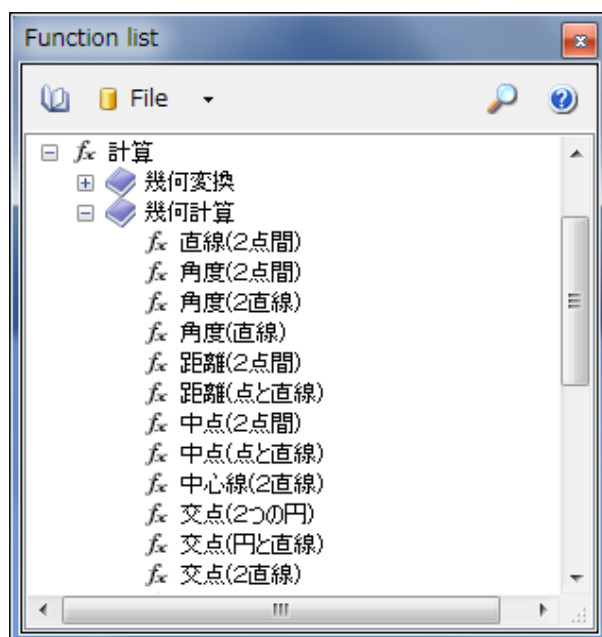
- ⑩ 「直線ハフデータ#2」の値を「直線の生成#2」の値とリンクします。

Name	Time (ms)	Reference
直線ハフデータ#2	0.016	
Result		直線ハフ検出#1.Datas
Index		
this		
a		
b		
c		
q		
Score		
直線の生成#1	0.003	
直線の生成#2	0.004	直線ハフデータ#2a 直線ハフデータ#2b 直線ハフデータ#2c
a		
b		
c		
this		

※直線の生成を組み入れるのは、「直線ハフデータ」の出力結果を直接、「交点(2直線)」へ渡せない為です。

直線ハフ検出の実行結果のデータ構造は、 $ax+by+c=0$ の直線式と q (角度)、 $score$ で構成されますが、交点(2直線)に渡すデータ構造、は $ax+by+c=0$ の直線式だけで構成されています。

- ⑪ Function list の「計算」、「幾何計算」の順にリストを展開し、「交点(2直線)」をダブルクリックします。



- ⑫ TASK タブのワークフローリストの最後の行に「交点(2直線)」が追加されます。

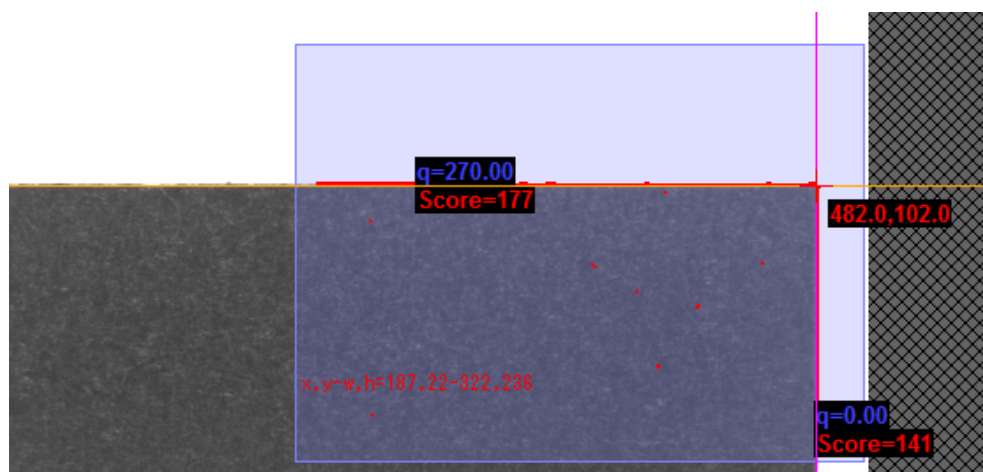
Name	Time (ms)	Reference
+ ファイル読み込み#1	0.811	
+ 2次元エッジ検出#1	7.486	+
+ 直線ハフ検出#1	0.876	
+ 直線ハフデータ#1	0.004	
+ 直線ハフデータ#2	0.016	
+ 直線の生成#1	0.003	
+ 直線の生成#2	0.004	
+ 交点(2直線)#1	0.009	

(FVIL, Caliper, Function, CrossPoint)

- ⑬ 実行アイコンをクリックしてみましょう。



- ⑭ 画像ビューに検出された交点が表示されます。



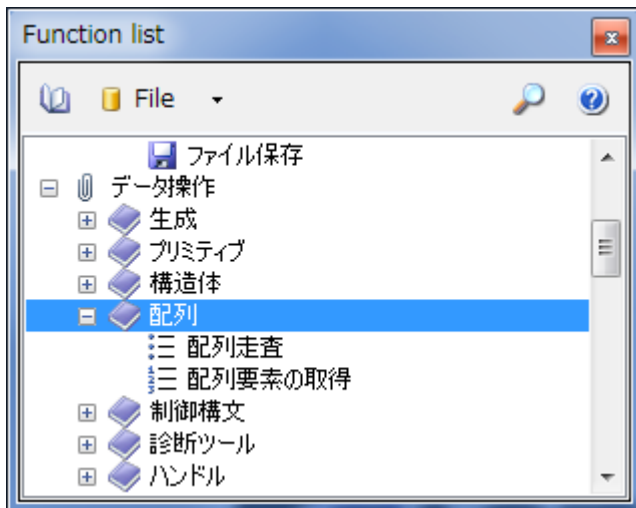
4. 応用編

WIL-Builder は配列走査や制御構文も行う事ができます。

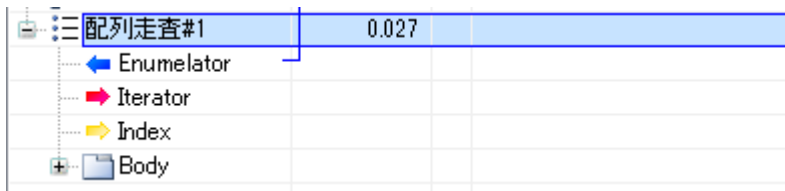
4.1 配列走査について

「配列走査」は、2 値ブローブ解析やグレイサーチなど複数個の結果を得られるデータから、個々のデータを使用するとき用います。個数分のループ処理を行います。

「配列走査」は、Function list では「データ操作」、「配列」の下にあります。



その構造は、以下のようにワークフローに表示されています。



Enumerator に配列走査したいデータをリンクし、Body の中で個々のデータ操作を記入します。

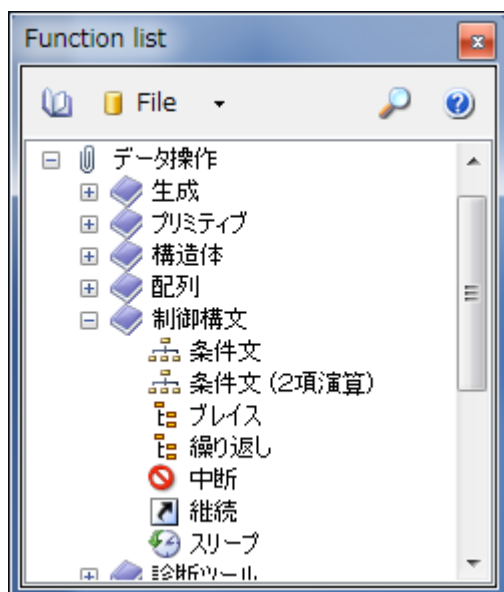
例としては Iterator に配列内のデータが入っており、それを Body の中のブローブデータへ渡して実行しています。

Name	Time (ms)	Reference
ファイル読み込み#1	1.699	
2値化#1	0.733	
2値ブローブ解析#1	0.367	+
配列走査#1	0.065	
Enumerator		2値ブローブ解析#1.BlobList
Iterator		
Index		
Body		
ブローブデータ#1	0.017	配列走査#1.Iterator
BlobList		
Index		
this		
Center		
Xdiff		
Ydiff		

左の点線内の Body 部分をデータ個数分ループ処理します。

4.2 ループ処理について

ループ処理を行う場合は、「データ操作」、「制御構文」の下に「繰り返し」を用意しています。
指定した回数分、繰り返し処理を行います。



その構造は、以下のようにワークフローに表示されています。

Name	Time (ms)	Reference
画像#1	0.002	
繰り返し#1	87.754	
Initial		
Loop		
Step		
Index		
Body		
時間計測...	0.000	
this		
平均化#1	0.185	
時間計測...	0.001	
Instance		時間計測(開始)#1.this
Time		
CSVファイル...	0.315	

body の部分が繰り返し実行されます。

上記の例では「フィルタ」の「平均化」を実行し、処理時間を CSV ファイルに書き込みます。

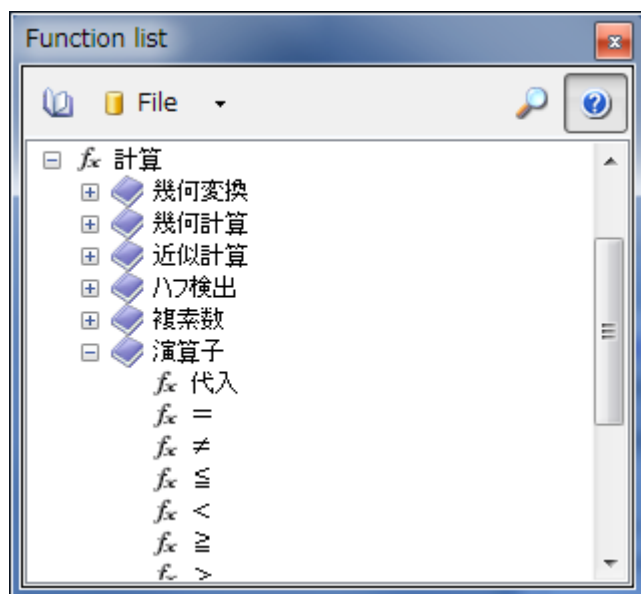
File	
Out	
Index	0
Parameter	
Initial	0
Loop	100
Step	1
Index	
for: 指標	

「Parameter」の「Index」を「Initial」の値で初期化し、1 回の実行ごとに「Step」の数だけ「Index」が加算されます。

「Index」が「Loop」で設定した値に達するまで、ループ処理を行います。

4.3 条件分岐について

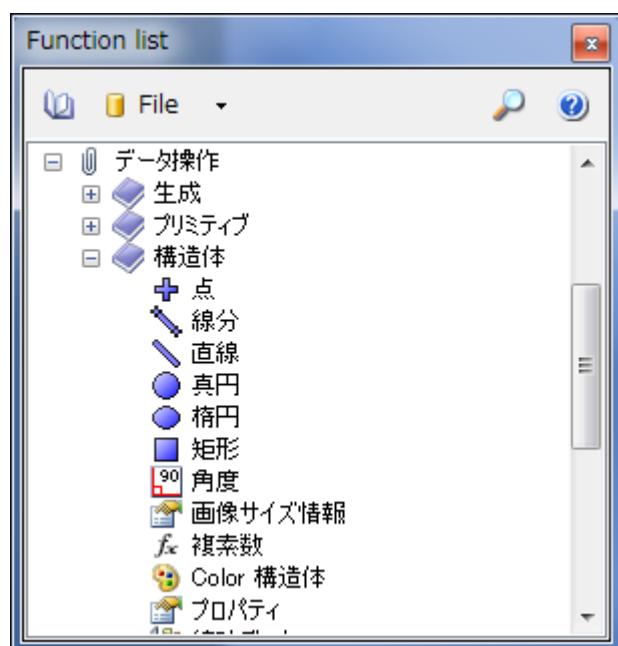
条件分岐を行う場合は、「計算」の「演算子」を選択し、その結果から処理を分岐することができます。「演算子」は、Function list では「計算」、「演算子」の下に各種用意しています。



条件分岐は、複雑なので、2 値ブローブ解析の結果による分岐を例にとり説明いたします。
なお、画像 2 値ブローブ解析までの説明は省略します。

- ① 「2 値ブローブ解析」の結果から有効ブローブ数 (EffectiveCount) を抽出するために「プロパティ」を追加します。

Function list の「データ操作」、「構造体」の順にリストを展開し、「プロパティ」をダブルクリックします。

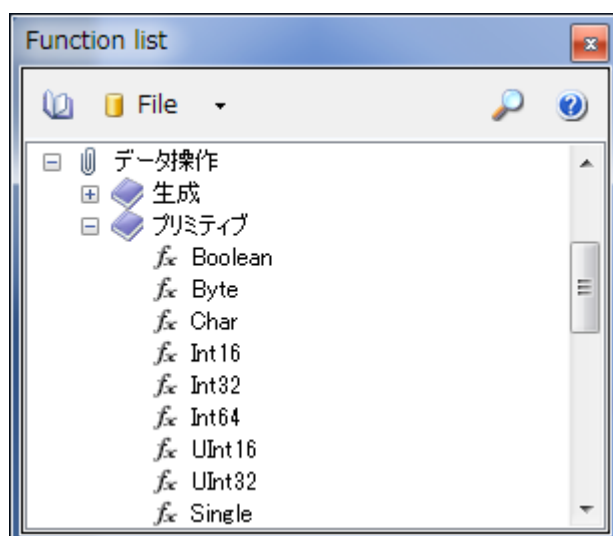


- ② TASK タブのワークフローリストの最後の行に「プロパティ」が追加されます。

Name	Time (ms)	Reference
画像#1	0.496	
2値化#1	0.337	
2値ブローブ解析#1	0.288	+
Src0		2値化#1.Dst0
BlobList		
BlobResult		
プロパティ#1	0.076	
In		2値ブローブ解析#1.BlobResult
EffectiveCount		

- ③ 次に条件分岐の対象となる値を設定するために「Int32」を追加します。

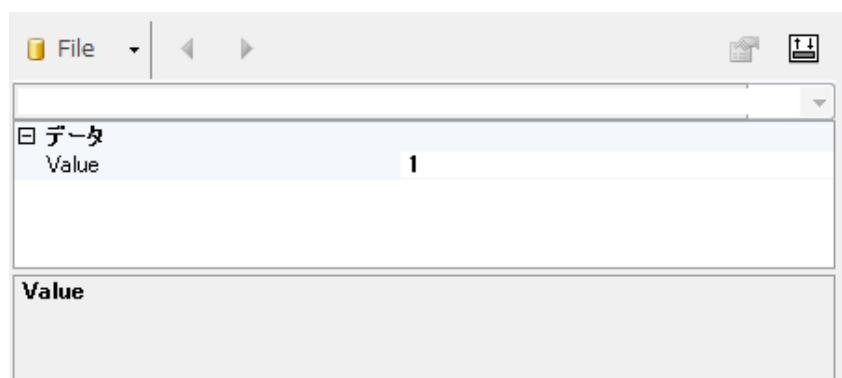
Function list の「データ操作」、「プリミティブ」の順にリストを展開し、「Int32」をダブルクリックします。



- ④ TASK タブのワークフローリストの最後の行に「Int32」が追加されます。

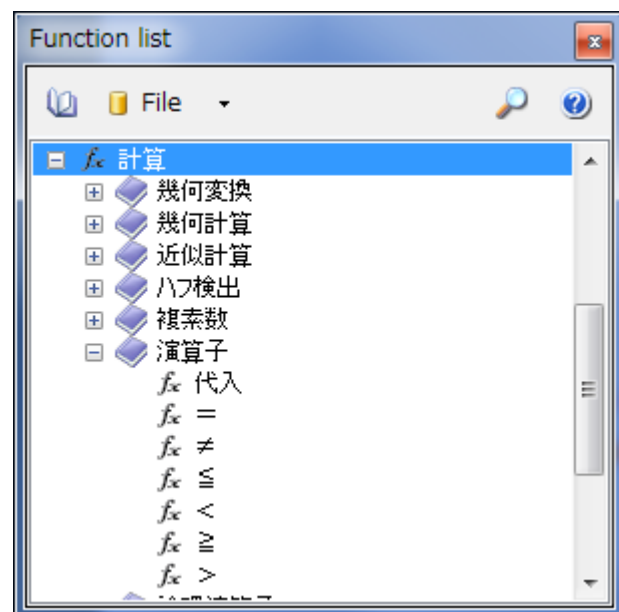
Name	Time (ms)	Reference
画像#1	0.028	
2値化#1	0.588	
2値ブローブ解析#1	3.095	+
プロパティ#1	0.196	
Int32#1	0.001	

- ⑤ 「Int32#1」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の「Value」を「1」に変更します。



- ⑥ 「2 値ブローブ解析」の結果から有効ブローブ数(EffectiveCount)と比較する値が設定できたので、条件を追加します。

Function list の「計算」、「演算子」の順にリストを展開し、「fx <」をダブルクリックします。



- ⑦ TASK タブのワークフローリストの最後の行に「fx <」が追加されます。

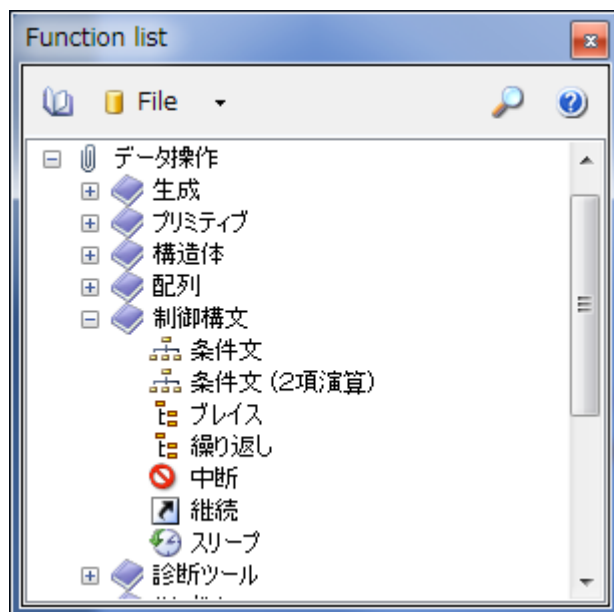
Name	Time (ms)	Reference
画像#1	0.028	
2値化#1	0.588	
2値ブローブ解析#1	3.095	+
プロパティ#1	0.196	
fx Int32#1	0.001	
fx < #1	0.004	

リンクは以下のようにになっています。

Name	Time (ms)	Reference
画像#1	0.028	
2値化#1	0.588	
2値ブローブ解析#1	3.095	+
プロパティ#1	0.196	
In		2値ブローブ解析#1.BlobResult
Out		
fx Int32#1	0.001	
fx < #1	0.004	
Left		Int32#1.this
Right		プロパティ#1.Out
Out		

- ⑧ 演算子の結果から条件文を追加します。

Function list の「データ操作」、「制御構文」の順にリストを展開し、「条件文」をダブルクリックします。



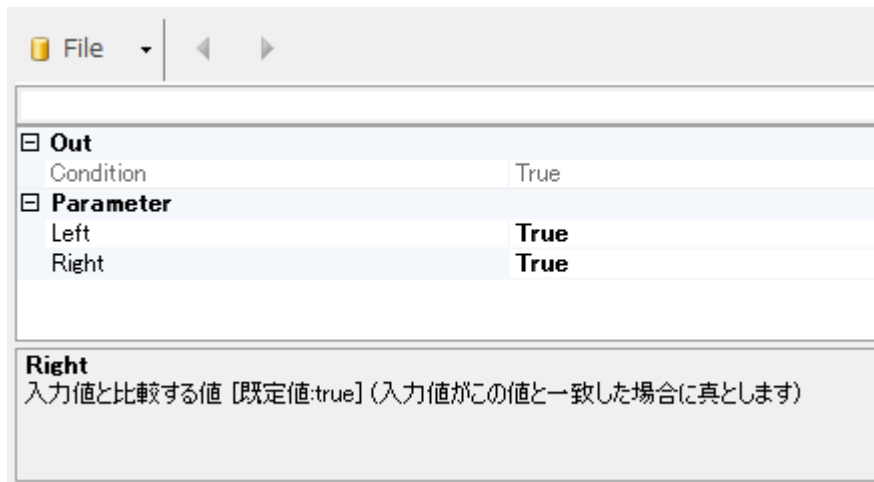
- ⑨ TASK タブのワークフローリストの最後の行に「条件文」が追加されます。さらにもう一つ「条件文」を追加します。

Name	Time (ms)	Reference
画像#1	0.003	
2値化#1	1.087	
2値ブローブ解析#1	0.086	+
プロパティ#1	0.018	
f_x Int32#1	0.000	
$f_x < \#1$	0.002	
条件文#1	0.000	
条件文#2	0.000	

- ⑩ 条件が真なら、body の部分を実行することが、可能です。もちろん偽の時に、body の部分を実行することも可能です。

Name	Time (ms)	Reference
画像#1	0.003	
2値化#1	0.177	
2値ブローブ解析#1	0.173	+
プロパティ#1	0.117	
f_x Int32#1	0.000	
$f_x < \#1$	0.003	
Left		Int32#1.this
Right		プロパティ#1.EffectiveCount
Out		
条件文#1	0.630	
Left		< #1.Out
Condition		
Body		

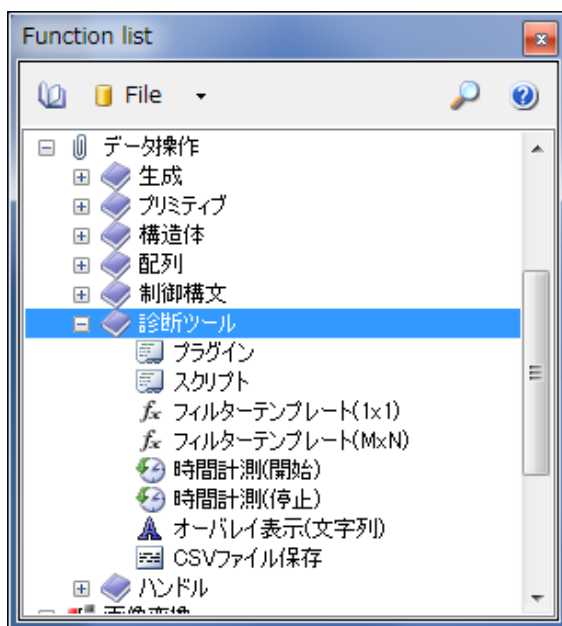
- ⑪ 「条件文」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Parameter の Left、Right を True から False にすることにより、真から偽へ変更する事ができます。



- ⑫ Body 部分に文字列を表示する機能を追加します。

Function list の「データ操作」、「診断ツール」の順にリストを展開し、「オーバーレイ表示(文字列)」をダブルクリックします。

さらにもう一つ「オーバーレイ表示(文字列)」を追加します。



- ⑬ TASK タブのワークフローリストの最後の行に「オーバーレイ表示(文字列)」が追加されます。
最後の行から「条件文#1」の body ヘドラッグします。
もう一つの「オーバーレイ表示(文字列)」を最後の行から「条件文#2」の body ヘドラッグします。

Name	Time (ms)	Reference
画像#1	0.003	
2値化#1	1.087	
2値ブローブ解析#1	0.086	+
プロパティ#1	0.018	
Int32#1	0.000	
<#1	0.002	
条件文#1	0.000	
Left		<#1.Out
Condition		
Body		
オーバーレイ表...	0.000	
In		
Position		
Out		
条件文#2	0.000	

- ⑭ 「条件文#1」の「オーバーレイ表示(文字列)」をクリックし、TASK タブの下側にプロパティグリッドが表示されていますので、Text に表示したい文字列を入力します。ここでは「NG」を入力します。
同様に「条件文#2」の「オーバーレイ表示(文字列)」には「OK」を入力します。

File			
Param			
Parameter			
Position	0, 0		
Text	NG		
TextPrefix			
TextSuffix			
Visualization			
Align	Top Left		
Text			
文字列化したデータ			

- ⑮ ワークフローは以下ようになります。

Name	Time (ms)	Reference
2値化#1	1.941	
2値ブローブ解析#1	0.111	+
プロパティ#1	0.015	
Int32#1	0.000	
<#1	0.003	
条件文#1	0.007	
Left		<#1.Out
Condition		
Body		
オーバーレイ表...	0.007	
条件文#2	0.000	
Left		条件文#1.Condition
Condition		
Body		
オーバーレイ表...	0.000	

- ⑯ 実行アイコンをクリックしてみましょう。



- ⑰ 画像ビューにブローブが検出された場合「NG」がモニタに表示されます。



4.4 プラグイン機能について

プラグイン機能とは、プログラミングでまとめた処理を作成し、WIL-Builder に読み込ませて処理することです。

もちろん、このまとめた処理はプログラムで使用することも可能です。

なお、次章で説明いたします「SAMPLE05」、「SAMPLE11」は、あらかじめプラグインを用意しています。

詳細は、「WIL-Builder 説明書」の「5.2.7 Plugin」をご参照ください。

5. サンプルワークフロー解説

WIL-Builder のサンプルワークフローについて、簡単に説明します。

5.1 SAMPLE00

SAMPLE00 を説明します。

機能 FPM 特徴点応用マッチング

内容 ファイルから読み込んだ画像の中から、あらかじめ登録していたパターンを FPM 特徴点応用マッチングを用いてサーチします。

TASK タブのワークフロー

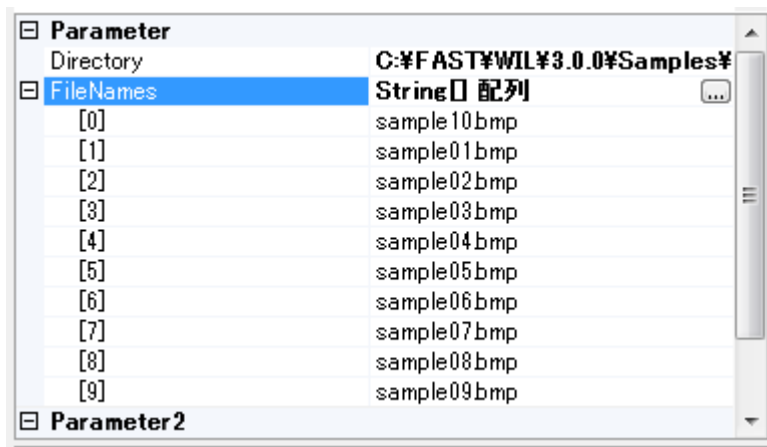
Name	Time (ms)	Reference
ファイル読み込み#1	0.914	
ファイル読み込み#2	1.640	
FPM2#1	6.074	+

POINT ファイル読み込み#1 について (FVIL. IFviFileAccess. Load)

プロパティグリッドの Parameter2 「FileIndexMode」が「Sequential」になっている場合、実行アイコンをクリック毎に読み込みファイルをインデックス順にロードします。



「FileNames」を展開するとファイル名が羅列されている。



5.2 SAMPLE01

SAMPLE01 を説明します。

機能 グレイサーチ

内容	ファイルから読み込んだ画像の中から、2つのマークそれぞれに対して正規化相関サーチを行い、2つの回答位置の間の距離を計測します。
----	---

TASK タブのワークフロー

Name	Time (ms)	Reference
ファイル読み込み#1	1.502	
ファイル読み込み#2	2.186	
ファイル読み込み#3	2.148	
GS2#1	0.744	+
GS2 データ#1	0.009	
GS2#2	1.043	+
GS2 データ#2	0.008	
距離(2点間)#1	0.022	

POINT 「GS2」のサーチ結果を「GS2 データ」から取得し、Point にリンクします。

Name	Time (ms)	Reference
ファイル読み込み#3	2.148	
GS2#1	0.744	+
GS2 データ#1	0.007	
GS2#2	1.043	+
GS2 データ#2	0.008	
距離(2点間)#1	0.022	
Point0		GS2 データ#1.Position
Point1		GS2 データ#2.Position
Distance		

5.3 SAMPLE02

SAMPLE02 を説明します。

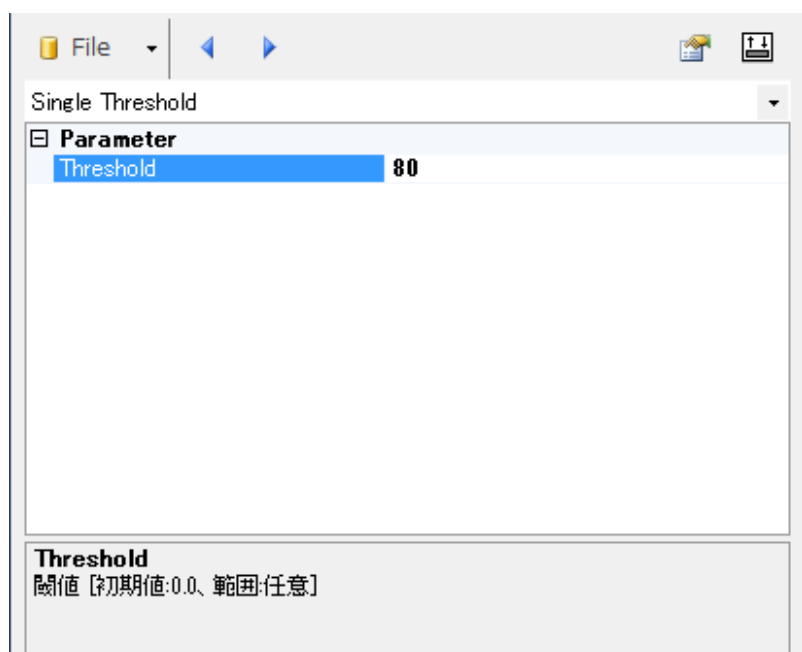
機能 2 値ブローブ解析

内容 ファイルから読み込んだ画像を固定 2 値化で 2 値化し、2 値ブローブ解析を実行し、ブローブを計測します。

TASK タブのワークフロー

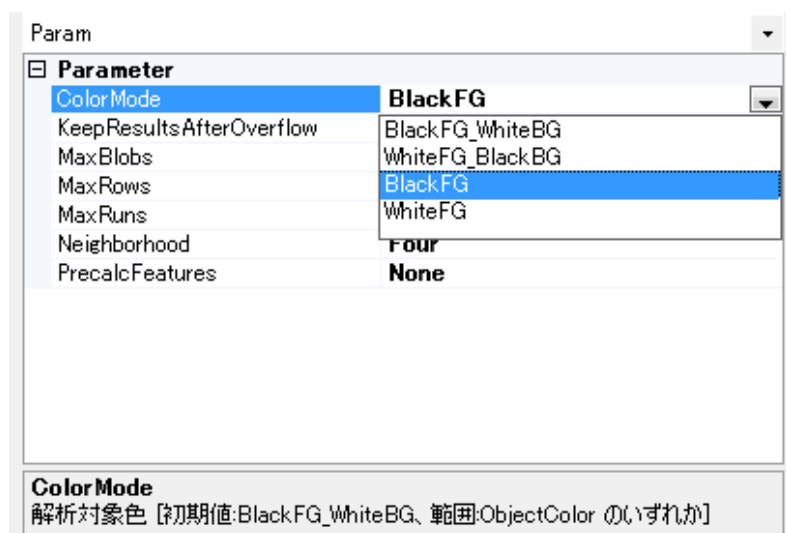
Name	Time (ms)	Reference
+ ファイル読み込み#1	3.290	
+ 2値化#1	107.450	
+ 2値ブローブ解析#1	0.138	+

POINT 「2 値化」により濃淡画像を白と黒の 2 値画像に変換します。



※対象ブローブが 2 値化できる最適なノードを選択してください。

POINT 「2 値ブローブ解析」の Parameter 「ColorMode」により、計測したい色を指定します。



5. 4 SAMPLE03

SAMPLE03 を説明します。

機能

コーナー座標計測 (R 計測)

内容

ファイルから読み込んだ画像から、指定した 2 か所の範囲から 1 次元エッジを検出し、求めた近似直線から交点を検出します。

TASK タブのワークフロー

Name	Time (ms)	Reference
ファイル読み込み#1	49.773	
1次元エッジ検出 (...)	14.838	+
1次元エッジ検出 (...)	0.282	+
直線近似#1	28.381	
直線近似#2	0.067	
交点(2直線)#1	1.344	

POINT 「1 次元エッジ検出」の検出領域は WOI を用いてマウスで設定できます。

Param

OUT

Result

Parameter

Around

Box

EdgeDirection

Foot

Len

LineWidth

MaxThreshold

Number

ScanDirection

Threshold

FVIL Edge.CFviEdgeResult

2

190.5, 329.5, 437, 407, 0, 0, 0

Both

5

0

0

75

17

Up

-1

POINT AUTO Link により、データの受け渡しが、図のように関連付けされています。

Name	Time (ms)	Reference
ファイル読み込み#1	49.773	
1次元エッジ検出 (...)	14.838	+
Src0		ファイル読み込み#1.Out
Box		
Result		
1次元エッジ検出 (...)	0.282	+
Src0		ファイル読み込み#1.Out
Box		
Result		
直線近似#1	28.381	
Points		1次元エッジ検出 (矩形)#1.Result
Line		
直線近似#2	0.067	
Points		1次元エッジ検出 (矩形)#2.Result
Line		
交点(2直線)#1	1.344	
Line0		直線近似#2.Line
Line1		直線近似#1.Line
Point		

5.5 SAMPLE04

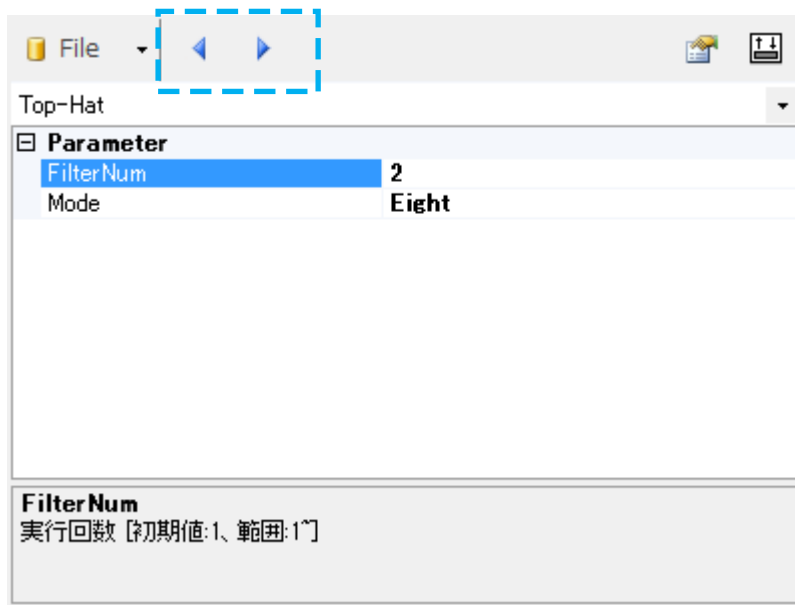
SAMPLE04 を説明します。

機能	モルフォロジを用いたキズ計測
内容	ファイルから読み込んだ画像からにモルフォロジを用いて対象物上のキズを消した後、元画像と画像間差分を行いキズのみ抽出します。明欠陥を抽出する場合は Top-Hat、暗欠陥を抽出する場合は Bottom-Hat を使用します。これらの他に minmax 差分を使用した方法もあります。

TASK タブのワークフロー

Name	Time (ms)	Reference
ファイル読み込み#1	1.886	
モルフォロジ (3x3)#1	31.465	
2値化#1	0.548	
2値ブローブ解析#1	0.102	+

POINT 「モルフォロジ」は複数の機能を持つノードがありますので、青いカーソルで機能の指標を切り替えることができます。青いカーソルは機能の指標をシーケンスに増減します。(左: 1つ減少、右: 1つ増加)



※最適なノードを選択してください。

5.7 SAMPLE06

SAMPLE06 を説明します。

機能 カラーサーチ

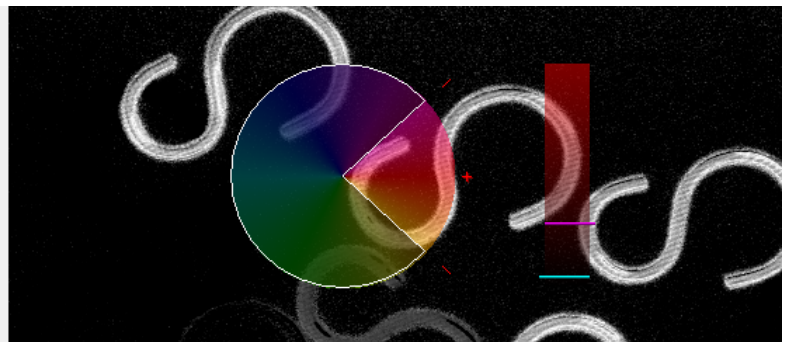
内容 ファイルから読み込んだ画像とパターン画像から濃淡画像に変換し、FPM 特徴点応用マッチングを用いてサーチします。

TASK タブのワークフロー

Name	Time (ms)	Reference
ファイル読み込み#1	43.957	
ファイル読み込み#2	4.333	
HSV濃淡化#1	6.444	
HSV濃淡化#2	0.695	
パターン登録#2	0.159	
FPM2#1	27.296	+

POINT 「HSV 濃淡化」Parameter である色相の基準や範囲、彩度の上下限はマウスで操作することが可能です。

Name	Time (ms)	Reference	Type
ファイル読み込み#1	2.952		FVILIFviFileAcc
ファイル読み込み#2	2.825		FVILIFviFileAcc
HSV濃淡化#1	6.444		FVILConversion
HSV濃淡化#2	0.695		FVILConversion
パターン登録#2	0.159		FVILDataCFviF
FPM2#1	141.092	+	FVILFPM2CFvi



5. 8 SAMPLE07

SAMPLE07 を説明します。

機能	2 値ブローブ解析からリージョン処理
内容	ファイルから読み込んだ 2 値画像から 2 値ブローブ解析を行い、その解析データをもとに、濃淡画像からリージョン処理を行い、濃淡画像から特徴量を計測します。 またブローブ毎の中心位置も表示します。

TASK タブのワークフロー

Name	Time (ms)	Reference
ファイル読み込み#1	30.111	
2値ブローブ解析#1	15.914	+
ファイル読み込み#2	40.369	
濃淡画像計測#1	40.781	+
配列走査#1	2.189	

POINT 「配列走査」に「2 値ブローブ解析」の結果を渡し、検出個数分ループします。
「Body」部分を個数分処理します。

Name	Time (ms)	Reference
ファイル読み込み#1	2.878	
2値ブローブ解析#1	0.356	+
Src0		ファイル読み込み#1.Out
BlobList		
BlobResult		
ファイル読み込み#2	8.083	
濃淡画像計測#1	1.886	+
配列走査#1	0.018	
Enumerator		2値ブローブ解析#1.BlobList
Iterator		
Index		
Body		
ブローブデー...	0.002	
BlobList		配列走査#1.Iterator
Index		
this		
Center		
Xdiff		
Ydiff		

5.9 SAMPLE08

SAMPLE08 を説明します。

機能 2 値ブローブ解析

内容 ファイルから読み込んだ画像を 2 値化し、2 値ブローブ解析で得られたブローブの重心から放射状に 1 次元エッジ検出し、その得られた座標から円近似を行います。

TASK タブのワークフロー

Name	Time (ms)	Reference
+ [Folder Icon] ファイル読み込...	2.338	
+ [Image Icon] 2値化#1	9.678	
+ [Image Icon] 2値ブローブ解...	1.290	+
+ [List Icon] 3 配列走査#1	8.219	

POINT 「配列走査」に「2 値ブローブ解析」の結果を渡し、1 次元エッジ検出を行い、そのエッジ群から円近似を行います。




Name	Time (ms)	Reference
+ [Folder Icon] ファイル読み込み#1	0.869	
+ [Image Icon] 2値化#1	2.895	
+ [Image Icon] 2値ブローブ解析#1	0.110	+
Src0		2値化#1.Dst0
BlobList		
BlobResult		
+ [List Icon] 3 配列走査#1	1.970	
Enumerator		2値ブローブ解析#1.BlobList
Iterator		
Index		
Body		
+ [Image Icon] ブローブデータ#1	0.011	
BlobList		配列走査#1.Iterator
Index		
this		
Center		
Xdiff		
Ydiff		
+ [Image Icon] 1次元エッジ検出 (放...	0.268	+
Src0		ファイル読み込み#1.Out
Radial.Center		ブローブデータ#1.Center
Radial.Radius		
Result		
+ [Image Icon] 円近似#1	0.108	

5. 10 SAMPLE09

SAMPLE09 を説明します。

機能	グレイサーチ
内容	ファイルから読み込んだ画像の中から、あらかじめ登録していたパターンをサーチします。

TASK タブのワークフロー

Name	Time (ms)	Reference
 ファイル読み込み#2	1.625	
 ファイル読み込み#1	0.904	
 GS2#1	78.972	+

5.11 SAMPLE10

SAMPLE10 を説明します。

機能 QR コード認識

内容 ファイルから読み込んだ画像の中から、QR コード認識を行い、2 個の QR コードの距離を計測します。

TASK タブのワークフロー

Name	Time (ms)	Reference
+ ファイル読み込み#1	3.284	
+ QRコード認識#1	94.335	+
+ QRコードデータ#1	0.046	
+ QRコードデータ#2	0.018	
+ 距離(2点間)#1	0.037	

POINT 「QR コードデータ」の Center を「距離(2点間)」へ渡し、2 点間の距離を算出します。

Center は Point 型なので、キャストは必要ありません。

Name	Time (ms)	Reference
+ ファイル読み込み#1	38.143	
+ QRコード認識#1	137.417	+
← Src0		ファイル読み込み#1.Out
→ Result		
+ QRコードデータ#1	0.095	
+ QRコードデータ#2	0.021	
← Result		QRコード認識#1.Result
← Index		
→ this		
→ Center		
→ Size		
→ Text		
→ MinCellWidth		
→ ApptsCount		
→ CellOnesCount		
→ DelimitingPtsCount		
+ 距離(2点間)#1	13.241	
← Point0		QRコードデータ#2.Center
← Point1		QRコードデータ#1.Center
→ Distance		



※同じフォルダの QR_SerialPort.bff は、カメラから画像を取り込み、QR コード認識を行い、デコードした文字列をシリアル通信で送信するサンプルです。

5. 12 SAMPLE11

SAMPLE11 を説明します。

機能	濃度投影によるピーク個数を出力
内容	入力画像の濃度投影を行い、その結果からピーク探索を行って、ピーク個数を出力します。濃度投影は水平方向のみに対応しています。また、このサンプルでは プラグイン (PeakDetection.dll) を使用しています。PeakDetection.dll は、処理結果のリストビュー表示を実装しています。 (FVIL.Parser.IParserNodeStatisticsCtrl)

TASK タブのワークフロー

Name	Time (ms)	Reference
 ファイル読み込...	23.202	
 PeakDetection...	44.323	+

POINT プラグイン「BoxCalc」内で、サーチ結果によりエッジ抽出の領域をオフセットする機能を実現しています。同じフォルダにある「sample05_nouse.bff」は、プラグイン「BoxCalc」を使用せず、Function だけで実現しています。

※あらかじめ SAMPLE11 と同じフォルダにある「PeakDetection.dll」はプラグインなので、データディレクトリ(マイ ドキュメントの WIL-Builder 3.0.0)にコピーしておく必要があります。

詳細は、「WIL-Builder 説明書」の「5.2.7 Plugin」をご参照ください。

6. 困ったときは

WIL のセットアップや動作確認で問題が発生した場合には以下の内容をご確認ください。ご確認いただいても解消されない場合は、ユーザ・サポートまでお問い合わせください。

6.1 トラブルシューティング

(1) WIL-Builder が起動できない

WIL-Builder、WILsample (C#)、WILsample (VB) 等、.NET Framework を使用したサンプルアプリケーションは .NET Framework 2.0 (SP2) がインストールされている必要があります。

また、弊社が提供する FVIL アセンブリを GAC (グローバルアセンブリキャッシュ) に登録する必要があります。

GACに関しては、リリースノート「GAC の登録と解除」の項をご参照ください。

6.2 ユーザ・サポートについて

弊社製品につきましてのお問い合わせはユーザ・サポートにて承ります。
ご質問の内容に下記の必要事項をお書き添えいただき、e-mail、FAX もしくは TEL にてお問い合わせください。

- ・ 御社名
- ・ 部署名
- ・ お名前
- ・ ユーザ登録番号
- ・ 機種名またはライブラリ名
- ・ システムおよびライブラリのバージョン

e-mail での問い合わせ：support@fast-corp.co.jp

FAX での問い合わせ：046-272-8692

TEL での問い合わせ：046-272-8691

受付時間：月曜日～金曜日（祝祭日および弊社指定の休日を除く）

9:00～12:00 13:00～17:00

◆ソースファイルの添付についてお願い◆

作成されたプログラムがハングアップしてしまう等のトラブルを抱えたお客様より、ソースファイルが添付された e-mail をいただくことがあります。まずは以下のような詳しい状況をご説明いただき、ユーザ・サポートまでご相談ください。サポートの過程で弊社にてソースファイルの解析が必要だと判断した場合には、ソースファイルの添付をお願いしております。

- ・ お使いのライブラリ製品の名称
- ・ お使いのシステム、ライブラリのバージョン
- ・ プログラムのどの部分（どのライブラリ）でハングアップするのか細かい情報
- ・ 問題の部分に関連のあるライブラリなどの戻り値は正常終了しているか
- ・ 発生頻度

WIL-Builder チュートリアル

2014 年 2 月 第 1 版発行

発行所 株式会社ファースト

本 社 〒242-0001 神奈川県大和市下鶴間 2791-5

ユーザ・サポート FAX 046-272-8692 TEL 046-272-8691
E-mail : support@fast-corp.co.jp

B-002799